Theory and Applications
of Natural Language Processing

Aims and Scope

The field of Natural Language Processing (NLP) has expanded explosively over the past decade: growing bodies of available data, novel fields of applications, emerging areas and new connections to neighboring fields have all led to increasing output and to diversification of research.

"Theory and Applications of Natural Language Processing" is a series of volumes dedicated to selected topics in NLP and Language Technology. It focuses on the most recent advances in all areas of the computational modeling and processing of speech and text across languages and domains. Due to the rapid pace of development, the diversity of approaches and application scenarios are scattered in an ever-growing mass of conference proceedings, making entry into the field difficult for both students and potential users. Volumes in the series facilitate this first step and can be used as a teaching aid, advanced-level information resource or a point of reference.

The series encourages the submission of research monographs, contributed volumes and surveys, lecture notes and textbooks covering research frontiers on all relevant topics, offering a platform for the rapid publication of cutting-edge research as well as for comprehensive monographs that cover the full range of research on specific problem areas.

The topics include applications of NLP techniques to gain insights into the use and functioning of language, as well as the use of language technology in applications that enable communication, knowledge management and discovery such as natural language generation, information retrieval, question-answering, machine translation, localization and related fields.

The books are available in printed and electronic (e-book) form:

* Downloadable on your PC, e-reader or iPad
* Enhanced by Electronic Supplementary Material, such as algorithms, demonstrations, software, images and videos
* Available online within an extensive network of academic and corporate R&D libraries worldwide
* Never out of print thanks to innovative print-on-demand services
* Competitively priced print editions for eBook customers thanks to MyCopy service http://www.springer.com/librarians/e-content/mycopy

For other titles published in this series, go to
www.springer.com/series/8899

Peter Spyns • Jan Odijk
Editors

Essential Speech and Language Technology for Dutch

Results by the STEVIN programme

Springer

Editors
Peter Spyns
Nederlandse Taalunie
The Hague
The Netherlands

Jan Odijk
UiL-OTS
University of Utrecht
Utrecht
The Netherlands

Foreword by
Linde van den Bosch
Nederlandse Taalunie
The Hague
The Netherlands

ISSN 2192-032X
ISBN 978-3-642-30909-0
DOI 10.1007/978-3-642-30910-6
Springer Heidelberg New York Dordrecht London

ISSN 2192-0338 (electronic)
ISBN 978-3-642-30910-6 (eBook)

Library of Congress Control Number: 2012955243

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

The STEVIN programme was not only an important scientific endeavour in the Low Countries, but also a quite rare case of a tight inter-institutional cross-border collaboration within the Dutch-speaking linguistic area. Four funding agencies, three ministerial departments and one intergovernmental organisation in Flanders and the Netherlands were involved in this programme. STEVIN is an excellent illustration of how a medium European language can set an example in the domain of language (technology) policy.

It remains extremely important that citizens can use their native language in all circumstances, including when they deal with modern ICT and leisure devices. For example, a very recent trend is that devices such as smart-phones and television sets become voice-controlled. But usually English speaking people are the first to benefit from such an evolution; other linguistic communities have to wait – some for ever? Not only does this pose a danger of reducing the overall functionality of a language (and an impoverishment of an entire culture), but also it threatens those groups in society that do not master the universal language. For example, elderly or disabled people, who deserve most to enjoy the blessings of modern technology, are in many cases the last ones to benefit from it. Therefore, R&D programmes that support the local language are needed. Also in the future, the Dutch Language Union will continue to emphasise this issue.

Many individuals have contributed to make STEVIN a success story, of all which I sincerely want to thank for their commitment. A particular mention goes to the funding government organisations from the Netherlands and Flanders.

I am confident that the STEVIN results will boost research in academia and technology development in industry so that the Dutch language can continue to "serve" its speakers well under all circumstances. Hence, it is with great pleasure that I invite you to discover the scientific results of the STEVIN programme.

The Hague, The Netherlands

Linde van den Bosch
General Secretary of the
Dutch Language Union (2004–2012)

Preface

Summarising a research programme that lasted for more than 6 years is a demanding task due to the wealth of deliverables, publications and final results of each of the projects concerned. In addition to the content-related topics, which interest scientists, research programmes also lead to new insights for policy makers and programme managers. The former want to discover advances in the state of the art, while the latter are eager to learn good practices in programme governance and management.

The STEVIN programme is no exception. In this work, the collaborators of each STEVIN R&D project have selected and summarised their scientific achievements. Even though the scientific accomplishments are the main focus of this volume, we have also added descriptions of some other particular aspects of the programme as a whole, such as its rationale, IPR management and the main conclusions of its final evaluation.

This volume is the result of a great deal of dedicated and hard work by many individuals, who, unfortunately, we cannot all mention by name as the list would be too long. We would first like to thank our colleagues of the Nederlandse Taalunie (NTU – Dutch Language Union), the members of the HLT steering board and the STEVIN programme office, the participants of the various STEVIN committees and related working groups, the project collaborators for their dedicated work and, of course, the funding organisations.

Additionally, we gratefully acknowledge everyone who has been involved in creating this volume. There are the authors of the various chapters. Also, the following members of the STEVIN international assessment panel (IAP) were so kind to, in addition to project proposals earlier on, review contributions to this volume as their last official duty for STEVIN:

- Gilles Adda – LIMSI (Paris)
- Nicoletta Calzolari – ILC (Pisa)
- Paul Heisterkamp – DaimlerChrysler (Ulm)
- Stelios Piperidis (& Sotiris Karabetsos) – ILSP (Athens)
- Gábor Prószéky – Morphologic (Budapest)

For this latter task, much-appreciated help came from the following internationally renowned researchers:

- Etienne Barnard (& Marelie Davel) – CSIR (Pretoria)
- Núria Bel – IULA (Barcelona)
- Nick Campbell – TCD (Dublin)
- Thierry Declerck – DFKI (Saarbrücken)
- Koenraad De Smedt – UIB (Bergen)
- Cédric Fairon – CENTAL (Louvain-la-Neuve)
- Steven Krauwer – UiL-OTS (Utrecht)
- Bente Maegaard – CST (Copenhague)
- Wim Peters (& Diana Maynard) – DCS-NLPG (Sheffield)
- Louis Pols – UvA (Amsterdam)
- Laurette Pretorius – UNISA (Pretoria)
- Steve Renals – ILCC (Edinburg)
- Justus Roux – CTEXT (Potchefstroom)
- Khalil Sima'an – UvA-ILLC (Amsterdam)
- Dan Tufiş – RACAI (Bucarest)
- Josef van Genabith – DCU (Dublin)
- Gerhard van Huyssteen – NWU (Potchefstroom)
- Werner Verhelst – ETRO-DSSP (Brussels)

Finally, we are also indebted to Springer-Verlag's editorial staff for their help, namely Dr. Olga Chiarcos and, in particular, Mrs. Federica Corradi Dell'Acqua.

It is our sincere hope and conviction that this volume will be of great interest to an international audience of researchers in human language technologies (HLT), in particular those who work on Dutch, to government officials active in HLT or language policy and to funders of science, technology and innovation programmes in general.

The STEVIN[1] programme was funded by the Flemish and Dutch governments (www.stevin-tst.org). Its results are presented at (www.stevin-tst.org/etalage) and are available via the HLT Agency (www.tst-centrale.org).

The Hague, The Netherlands — Peter Spyns
STEVIN programme coordinator

Utrecht, The Netherlands — Jan Odijk
Chair of the STEVIN programme committee

[1] STEVIN stands for 'Essential Speech and Language Technology Resources'. In addition, Simon Stevin was a seventeenth century applied scientist who, amongst other things, introduced Dutch terms for mathematics and physics concepts. He worked both in Flanders and the Netherlands. Hence, his name is a perfect acronym for this joint programme. And he became famous for building a land yacht for Prince Maurice of Orange.

Acknowledgements

All projects reported on in this volume were funded by the STEVIN programme. STEVIN was organised under the auspices of the Dutch Language Union and jointly financed by the Flemish and Dutch governments. The Flemish government was represented by the Department of Economy, Science and Innovation (EWI), the Agency for Innovation (IWT), and the Research Foundation – Flanders (FWO). The Dutch government was represented by the Ministry for Economy, Agriculture and Innovation (E,L&I), the Ministry of Education, Culture and Science (OCW), the Netherlands Organisation for Research (NWO) and the Agency NL.

Contents

Chapter 1
Introduction

Peter Spyns

1.1 Context

The STEVIN ("STEVIN" is a Dutch acronym for "Essential Speech and Language Technology Resources for Dutch") programme aimed to contribute to the further progress of Human Language Technology for Dutch (HLTD) in the Low Countries (i.e., Flanders and the Netherlands) and to stimulate innovation in this sector. The major scientific goals were to set up an effective digital language infrastructure for Dutch, and to carry out strategic research in the field of language and speech technology for Dutch.[1] Consortia could submit project proposals in response to calls for proposals. Several calls were issued, and they included three open calls and two calls for tender as well. The thematic priorities for each call were determined in line with the overall STEVIN priorities and the state of their realisation before each call. The STEVIN thematic priorities, based on what is called the Basic Language Resource Kit (BLARK) for Dutch [20], are summarised in Tables 1.1 and 1.2. A BLARK is defined as the set of basic HLT resources that should be available for both academia and industry [13].

STEVIN advocated an integrated approach: develop text and speech resources and tools, stimulate innovative strategic and application-oriented research, promote embedding of HLT in existing applications and services, stimulate HLT demand via

[1] We refer the reader to Chap. 2 for more details.

P. Spyns (✉)
Nederlandse Taalunie, Lange Voorhout 19, 2514 EB Den Haag, Nederland
e-mail: pspyns@taalunie.org

Vlaamse overheid – Departement Economie, Wetenschap en Innovatie, Koning Albert II-laan 35, bus 10, B-1030 Brussel, België
e-mail: Peter.Spyns@ewi.vlaanderen.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_1, © The Author(s) 2013

Table 1.1 Summary of STEVIN scientific priorities – resources and research

Speech	Resources (I)	Strategic research (II)
	1. Speech and multimodal corpora for:	1. Robustness of speech recognition
	1. (a) Computer-Assisted Language Learning applications	
	1. (b) Applications in which names and addresses play an important role	
	1. (c) Call centres question and answer applications	
	1. (d) Educational applications	
	2. Multimodal corpora for applications of broadcast news transcription or person identification	2. Output treatment (inverse text normalisation)
	3. Text corpora for the development of stochastic language models	3. Confidence measures
	4. Tools and data for the development of:	4. Adaptation
	4. (a) Robust speech recognition	5. Lattices
	4. (b) Automatic annotation of corpora	
	4. (c) Speech synthesis	
Text	Resources (IV)	Strategic research (V)
	1. Richly annotated monolingual Dutch corpora	1. Semantic analysis, including semantic tagging and integrating morphological, syntactic and semantic modules
	2. Electronic lexicons	2. Text pre-processing
	3. Aligned parallel corpora	3. Morphological analysis (compounding and derivation)
		4. Syntactic analysis (robust parsing)

Table 1.2 Summary of STEVIN scientific priorities – application oriented

	Embedding HLTD
Speech (III)	1. Information extraction from audio transcripts created by speech recognisers
	2. Speaker accent and identity detection
	3. Dialogue systems and Q&A solutions, especially in multimodal domains
Text (VI)	1. Monolingual or multilingual information extraction
	2. Semantic web
	3. Automatic summarisation and text generation applications
	4. Machine translation
	5. Educational systems

demonstration projects and encourage cooperation and knowledge transfer between academia and industry. As all these aspects were targeted in one and the same programme, the structure and goals of STEVIN were based on the theoretical notion of a *stratified innovation system*. The main idea behind a stratified innovation

Table 1.3 Distribution of the STEVIN scientific projects (=the HLTD supply side) over the layers of a stratified innovation system – demonstrators, representing the application layer (=the HLTD demand side), are not included

	BLARK for Dutch	HLTD R&D	Embedding HLTD
Speech	Autonomata	Autonomata TOO	
	SPRAAK	MIDAS	DISCO
	STEVINcanPRAAT		
	JASMIN-CGN	NBest	
Text	D-Coi		
	LASSY		DuOMAn
	COREA	DAESO	Daisy
	Cornetto		
	DPC	PACO-MT	
	IRME		
	SoNaR		

system is that the strata or layers of an innovation system[2] do not exist in isolation, but build on one another [1, p. 63]. Consequently, each layer requires a proper set of government support measures that nevertheless have to be consistent and reinforce one another. For example, the STEVIN third open call, which focussed on application oriented projects, favoured proposals that used results (basic resources) of earlier STEVIN projects.

Modern theory on innovation systems states that no central entity can "steer" a sector or domain, but that knowledge (not limited to scientific expertise but also legal knowledge, business expertise etc.) of the sector or domain is distributed over the various actors in an innovation system. Hence, interactions and connections between these (different types of) actors had to be considered as well. Therefore, in addition to scientific projects, STEVIN also funded networking and dissemination activities. Depending on the focus of the projects, they are situated in a different layer of the innovation system. Table 1.3 shows all the STEVIN scientific projects (cf. Sect. 1.2.1) situated in the appropriate layer of a stratified innovation system. Four layers are distinguished:

- "BLARK": create basic HLT for Dutch (HLTD) resources – e.g., compile a large annotated corpus of written Dutch;
- "R&D": perform HLTD research[3] – e.g., investigate methods and build components that enhance the noise robustness of a speech recogniser;
- "embedding": enhance the functionality of applications thanks to HLTD – e.g., integrate a speech component in a computer-assisted language learning (CALL) system;
- "applications": create end-user HLTD applications – e.g., a speech therapy application for persons with a cochlear implant.

[2]Cf. the column labels of Table 1.3 and see [2] for the definition of an innovation system.

[3]In the case of STEVIN, it concerned strategic research, not fundamentel research.

In total, 14 demonstrators (cf. Sect. 1.2.2) were built mainly by small and medium sized enterprises (SMEs) (and hence represent the HLTD demand side), while 19 scientific projects (cf. Sect. 1.2.1) were carried out mainly by academic partners (the HLTD supply side).

1.2 STEVIN Projects

The most salient results of the various STEVIN projects are summarised below. Section 1.2.1 contains the main results of the scientific projects. In order to be complete, we enlist in Sect. 1.2.2 the other STEVIN projects as well. As their main goal was rather to create visibility for HLTD in the Low Countries than to achieve scientific progress, these projects are not further described in this volume.

1.2.1 STEVIN Scientific Projects

The numbers of the enumeration also refer to Fig. 1.1 of Sect. 1.3.2.

1. Autonomata (*Automata for deriving phoneme transcriptions of Dutch and Flemish names* – cf. Chap. 4) built two resources: (1) a grapheme-to-phoneme (g2p) conversion tool set for creating good phonetic transcriptions for text-to-speech (TTS) and automatic speech recognition (ASR) applications with a focus on phonetic transcriptions of names [27], and (2) a corpus of 72,000 spoken name utterances supplied with an orthographic and auditorily verified phonetic transcription [26]. These resources have been used in the Autonomata TOO project (cf. project 12).
2. SPRAAK (*Speech Processing, Recognition and Automatic Annotation Kit* – cf. Chap. 6) re-implemented and modernised a speech recognition tool kit and provided demo recognisers for Dutch. The SPRAAK tool kit combines many of the recent advances in automatic speech recognition with a very efficient decoder in a proven hidden Markov model (HMM) architecture (cf. project B in Fig. 1.1) [8]. SPRAAK is flexible modular tool kit meant for speech recognition research and a state of the art recogniser with an extensive programming interface.
3. STEVINcanPRAAT (cf. Chap. 5) extended the functionality of the widely used PRAAT open source package for doing phonetics by computer (cf. project A) [3]. In particular a Klatt synthesiser, a vowel editor and some under the hood improvements were added to the PRAAT system. The updated software is freely available via the regular PRAAT distribution channel (www.praat.org).
4. JASMIN-CGN *(Extension of the CGN with speech of children, non-natives, elderly and human-machine interaction)* – cf. Chap. 3 extended the Spoken Dutch Corpus (CGN – cf. project A in Fig. 1.1) with 115 h of read speech

and human-machine dialogues of children, non-natives and elderly people in the Netherlands (2/3) and Flanders (1/3). All recordings were delivered with a verbatim orthographic transcription, a transcription of the human-machine interaction (HMI) phenomena, part of speech (POS) tagging and an automatic phonetic transcription [6].

5. D-Coi (*Dutch Language Corpus Initiative* – cf. Chap. 13) was a preparatory project that created a blueprint for the construction of a 500-million-word corpus of contemporary written Dutch (SoNaR – cf. project 11) [16]. A set of annotation protocols and other reports useful for corpus building have been made available. A 54-million-word pilot corpus was compiled, parts of which were enriched with linguistic annotations. The corpus exploitation tool of the CGN (cf. project A) was adapted to cope with written text data.
6. LASSY (*LArge Scale SYntactic annotation of written Dutch* – cf. Chap. 9) created a large one-million-word corpus of written Dutch texts (LASSY small) that was syntactically annotated and manually corrected [23]. In addition, a 1.5-billion-word corpus (LASSY Large) was annotated automatically with part-of-speech and syntactic dependency information. Various browse and search tools for syntactically annotated corpora as well as the Alpino parser (cf. project D in Fig. 1.1) [24] were extended. These were used by DPC (cf. project 9) and SoNaR (cf. project 11).
7. COREA (*COreference Resolution for Extracting Answers* – cf. Chap. 7) implemented a robust tool to resolve coreferential relations in text and to support annotation activities by humans [11]. It is relevant for a range of applications, such as information extraction, question answering and summarisation. A corpus (in Dutch) was annotated with coreferential relations of over 200,000 words. In addition, general guidelines for co-reference annotation are available.
8. Cornetto (*Combinatorial and Relational Network as Tool Kit for Dutch Language Technology* – cf. Chap. 10) built a lexical semantic database for Dutch by combining and aligning the Dutch WordNet and the Reference File Dutch (Referentiebestand Nederlands). It includes the most generic and central part of the Dutch vocabulary and a specialised database for the legal and finance domains [31]. In total the Cornetto database contains more than 70,000 concepts, 92,000 words and 120,000 word meanings. Also a tool kit for the acquisition of new concepts and relations was implemented. This tool kit facilitates the tuning and extraction of domain specific sub-lexica from a compiled corpus. It was used in e.g., the FP7 Kyoto project [30].
9. DPC (*Dutch Parallel Corpus* – cf. Chap. 11) is a ten-million-word parallel corpus comprising texts in Dutch, English and French with Dutch as central language [17]. It consists of two sentence-aligned bilingual corpora (Dutch-English and Dutch-French) with a portion aligned at a sub-sentential level as well. The corpus has four translation directions (at least two million words per direction) and is a balanced corpus including five text types. A user friendly interface (parallel web concordancer) to query the parallel corpus is available on-line.

10. IRME (*Identification and Representation of Multi-word Expressions* – cf. Chap. 12) carried out research into sophisticated methods for automatically identifying MWEs in large text corpora and into a maximally theory-neutral lexical representation of MWEs. With an identification method derived from the research, a list of MWEs and their properties were automatically identified and formed the basis for the corpus-based DuELME Dutch lexical database of MWEs [10]. This DuELME database was later (not in the STEVIN context) adapted to be compliant with the Lexical Mark-up Framework (LMF).
11. SoNaR (*STEVIN reference corpus for Dutch* – cf. Chap. 13) constructed a 500-million-word reference corpus of contemporary written Dutch texts of various styles, genres and sources. The entire corpus was automatically tagged with parts of speech (POS) and lemmatised. In addition, for a one-million-word subset of the corpus different types of semantic annotation were provided, viz. named entity labels, co-reference relations, semantic roles and spatial and temporal relations. Tools and materials from other STEVIN projects (D-Coi, LASSY, COREA – cf. projects 5–7 respectively) were re-used. An important aspect of the project consisted of clearing the IPR for the various types of corpus material and documenting the acquisition process [19].
12. Autonomata TOO (*Autonomata Transfer of Output* – cf. Chap. 14) tackled the problem of spoken name recognition in the context of an automated Point of Interest (POI) providing business services [18]. New solutions were found by exploiting and extending the phoneme-to-phoneme (p2p) learning tools that were developed in the Autonomata project (cf. project 1). Autonomata Too delivered a demonstrator of a POI providing service and p2p converters for POI name transcription. Furthermore, it produced a corpus of read-aloud POI names from Belgium and the Netherlands. This corpus consists of 5,677 sound files and corresponding manually created phonetic transcriptions.
13. MIDAS (*MIssing DAta Solutions* – cf. Chap. 16) tackled the noise robustness problem in automatic speech recognition by missing data techniques, which enables masking out "unreliable" parts of the speech signal (due to noise etc.) during the recognition process [9]. The missing information is reconstructed by exploiting the redundancy in the speech signal. The algorithms were implemented and integrated in the SPRAAK speech recognition tool kit (cf. project 2).
14. NBest (*Dutch Benchmark Evaluation of Speech Recognition Technology* – cf. Chap. 15) developed an evaluation benchmark for large vocabulary continuous speech recognition in Dutch as spoken in Flanders and the Netherlands. It defined four primary tasks based on transcriptions of broadcast news and conversational telephony style speech in Northern and Southern Dutch [12]. The project defined evaluation protocols and training material, and collected evaluation data sets. Seven academic speech recognition systems – including SPRAAK (cf. project 2) – participated in the benchmark evaluation [28].
15. DAESO (*Detecting And Exploiting Semantic Overlap* – cf. Chap. 8) implemented tools for the automatic alignment and classification of semantic relations (between words, phrases and sentences) for Dutch, as well as for a Dutch

text-to-text generation application that fuses related sentences into a single grammatical sentence. The project also built a two-million-word monolingual parallel corpus [14]. In addition, three specific corpus exploitation tools were implemented as well as a multi-document summariser for Dutch.

16. PACO-MT (*Parse and Corpus based Machine Translation* – cf. Chap. 17) built a hybrid machine translation system for Dutch-English and Dutch-French (in both directions) integrating linguistic analysis and a transfer component based on syntactic structures into a data-driven approach [29]. Some specific components were implemented, such as a node aligner, a grammar rule inducer, a decoder and a target language generator. In addition, more than 48 resp. 45 million source words of parallel texts for Dutch-English resp. Dutch-French were collected.
17. DISCO (*Development and Integration of Speech technology into COurseware for language learning* – cf. Chap. 18) developed an ASR-based Computer-Assisted Language Learning (CALL) prototype for training oral proficiency for Dutch as a second language (DL2). The application optimises learning through interaction in realistic communication situations and provides intelligent feedback on various aspects of DL2 speaking, viz. pronunciation, morphology and syntax [21]. It uses the SPRAAK tool kit – cf. project 2.
18. DuOMAn (*Dutch Online Media Analysis* – cf. Chap. 20) developed a set of Dutch language resources (including sentiment-oriented lexica) and tools for identifying and aggregating sentiments in on-line data sources [22]. The tools support automated sentiment analysis, parsing, entity detection and co-reference resolution with an emphasis on robustness and adaptability. An on-line demonstrator is available.
19. Daisy (*Dutch lAnguage Investigation of Summarisation technologY* – cf. Chap. 19) developed and evaluated technology for automatic summarisation of Dutch informative texts. Innovative algorithms for topic salience detection, topic discrimination, rhetorical classification of content, sentence compression and text generation were implemented [7]. A demonstrator was built and the Alpino parser (cf. project D in Fig. 1.1) was extended with a text generation and fluency restoring component. In addition, a tool that segments and classifies the content of Web pages according to their rhetorical role was implemented.

1.2.2 Other STEVIN Projects

The "other" projects mainly include demonstration projects. They served to convincingly illustrate the feasibility of applying HLT in end-user applications and services in Dutch. The purpose was to stimulate the uptake of HLTD by industry. Therefore, the main applicant had to be a private company. Two "educational projects" had to increase the interest of students for HLT and HLT related studies. Two master classes targeted decision makers in industry and government to increase their awareness of the potentialities of adopting HLT in their organisation. Four

of these "other" projects (labelled i–iv) are included in Fig. 1.1 as they build on resources of earlier STEVIN scientific projects. We refer the interested reader to the "STEVIN programme: project results" booklet[4] for more detailed descriptions of these projects.

1. The "*licence plate line*" ("Kentekenlijn") allows Dutch police officers on the road (in a car, on a bicycle, on foot) to check registration plates of vehicles in a hands-free (and eyes-free) manner using the NATO alphabet. Various types of information on the car are read aloud using speech synthesis. As a result, fewer officers are needed in the police control room to manage this type of calls and requests. Hence, they can spend more time on more urgent and higher priority calls. And, more requests for licence plate numbers can be processed.
2. The Dutch information portal for legal professionals ("*Rechtsorde*") provides a more user-friendly access to information about laws and local government regulations as available in official legal publications. The system corrects spelling errors and suggests synonyms and closely related terms based on compound decomposition and inflectional analysis.
3. *"CommuneConnect!"* ("GemeenteConnect") is a phone dialogue system that allows for free speech input that provides the caller with information on legislation and procedures that apply in a commune (question-answering). It uses a combination of state-of-the-art speech recognition, classification and computational linguistics based dialogue management.
4. A *spell check chatbot* ("Spelspiek") provides the correct spelling of pseudo-phonetically spelled words that present spelling difficulties. If several spelling alternatives exist, extra explanation is added. It consists of an automatic conversational agent that behaves as an expert in Dutch spelling. The core of the system consists of a one-million-word vocabulary, a spelling error database and smart spelling error recognition algorithms. The chatbot also knows how to respond to "unexpected" input by exhibiting some sense of humour. Currently, the service is available through a webpage and Twitter.
5. "*SonaLing*" ("Klinkende Taal") is a dynamic jargon detection system to avoid administrative and complicated language in documents and written customer communication by local governments. It focusses on readability and revision advice. The project resulted in several commercial product offers, including a freely accessible web tool, and a large user base.
6. *WebAssess* allows for the automatic pre-selection of call centre agent candidates during the recruitment process. The total set-up includes an e-learning module via the internet and a speech interactive voice response system that mimics a customer calling a contact centre. The system checks the replies of the candidate on the presence of need-to-have-answers and nice-to-have-answers, and adapts the dialogue flow accordingly.

[4] http://www.stevin-tst.org/english/

7. *Primus* adapted Microsoft's Dutch spelling and grammar checking tools for use by dyslectic users. Adapted grammar rules provide more useful correction suggestions and a text-to-speech module pronounces the suggestions.
8. A Flemish editor offers a daily audio edition "*Audio Newspaper*" ("Audiokrant") for visually disabled persons of two popular newspapers. A daily production process using speech synthesis generates CDs that contain a complete spoken newspaper. The CDs are compliant with the international DAISY (digital accessible information system) standard that allows for navigation over the newspaper content.
9. The "*NeOn*" project (labelled "iii" in Fig. 1.1) combines speech segmentation, speech recognition, text alignment and sentence condensation techniques to implement a less labour intensive semi-automatic tool to produce Dutch subtitles for certain television shows (for the Dutch NPO and Flemish VRT broadcasting organisations). This resulted in a reduction of up to 40 % in processing time compared to the method traditionally used. A follow-up project has been initiated by the VRT.
10. The "*Justice recognition*" ("Rechtspraakherkenning") application produces transcriptions of recordings made in the courtrooms in the Netherlands. The recordings are made searchable to enable retrieval of relevant items from the fully recorded lawsuits. In addition, a spoken summary of the trial can be generated. Even if the transcriptions are not completely accurate, the application significantly reduces the human effort in producing full transcriptions. Also the search and retrieval function is well appreciated. Several governmental organisations in the Netherlands have shown interest in this application.
11. *Woody* is a self-correcting talking word prediction system built for dyslectic users. Word lists and word prediction algorithms form the core of the tool. The project was the basis for a subsequent commercially available product called Wody.
12. The *"literacy plan foreign speakers* ("Alfabetisering Anderstaligen Plan" or AAP – labelled "ii" in Fig. 1.1) demonstrator can be used to train knowledge of the Dutch language, pronunciation of Dutch, and Dutch literacy. It uses speech recognition software and can be integrated in an existing language learning application for second language learners with a very limited level of literacy and limited knowledge of Dutch to produce speech feedback.
13. The *Hatci* project (labelled "iv" in Fig. 1.1) resulted in an automatic speech assessment system that can support a speech therapist in helping a patient with a cochlear implant to learn to speak. The tool plays an audio file (and/or a video file to allow for lip reading) to prompt a patient. A speech recogniser analyses the accuracy of the reproduction by the patient and hence assesses his/her hearing and speech reproduction abilities.
14. The "*YourNews*" news brokerage service uses language technology to collect, summarise and classify more than 1,000 newspaper articles per minute in accordance with the International Press and Telecom Council (ITPC) classification standard.

15. Two master classes were prepared and organised: one on ICT and dyslexia, and a second one on a general introduction on HLT for decision makers of mainly public organisations.
16. *Kennislink* is a website popular in the Low Countries mainly used by students and teachers to find information about recent scientific developments. Ninety-three articles on STEVIN projects and HLT in general were added to the Kennislink website. In addition, two perception studies were organised amongst students: one to rate the Kennislink HLT articles and one about their familiarity with and interest in HLT.
17. The *DiaDemo* "educational" application (labelled "i" in Fig. 1.1) can detect on the spot to which main dialect group a Flemish person belongs on basis of a few utterances.

1.3 Mission Accomplished

1.3.1 Addressing the STEVIN Priorities

To know the extent to which STEVIN has achieved its targets as defined at the start of the programme, the STEVIN priorities are compared to the topics and output of the various projects. Table 1.4 shows the distribution of the 19 scientific projects (cf. Sect. 1.2.1 for their short descriptions) over the STEVIN priorities as detailed in Tables 1.1 and 1.2 (cf. Sect. 1.1). The subsequent chapters of this volume provide ample details of each STEVIN scientific project.

The SPRAAK project, in combination with the MIDAS project, covered the development of a robust speech recogniser with additional features for noise robustness (priorities II.1–5). SPRAAK is re-used by the DISCO project that itself is a computer-assisted language learning application (priority VI.5). Autonomata and Autonomata TOO address the issues regarding the correct synthesis (priority I.4.c) and robuster recognition (priority II.1) of proper nouns, street names and names of points of interest (priority I.1.b), which is highly relevant for (car) navigation systems and call centre applications (priorities I.1.c). STEVINcanPRAAT basically is an improvement of the PRAAT (phonetic) signal processing tool (priority I.4.c). The JASMIN-CGN project extended the already available Spoken Dutch Corpus, in a manner useful for CALL applications (priorities I.1.a and I.1.d), and built automatic speech corpus annotation tools (priorities I.4.a–b).

Many STEVIN scientific projects obviously dealt with the creation and annotation of a corpus for written Dutch: D-Coi, LASSY, IRME and, of course, SoNaR that built a reference corpus of written Dutch of 500 million words (priorities IV.1 and I.3). The SoNaR corpus was annotated automatically using pre-processing tools and syntactico-semantic annotation tools and tagging schemas resulting from the D-Coi corpus pilot project (priorities IV.1). Also the COREA co-reference tools were used to annotate the SoNaR corpus. Lexica (priorities IV.2) were built by the IRME, Cornetto and DuOMAn projects. The DAESO tools focused on alignment

Table 1.4 STEVIN scientific projects mapped on the STEVIN priorities (cf. Tables 1.1 and 1.2) they mainly address – empty cells represent priorities not covered

Speech	Resources (I)	Strategic research (II)	Applications (III)
	1.(a) JASMIN-CGN	1. Autonomata TOO, SPRAAK, MIDAS	1.
	1.(b) Autonomata, Autonomata TOO		
	1.(c) Autonomata, Autonomata TOO		
	1.(d) JASMIN-CGN		
	2.	2. SPRAAK	2.
	3. JASMIN-CGN, Autonomata	3. SPRAAK, MIDAS	3.
	4.(a) JASMIN-CGN	4. SPRAAK	
	4.(b) JASMIN-CGN	5. SPRAAK	
	4.(c) Autonomata, Autonomata TOO, (STEVINcanPRAAT)		
Text	Resources (IV)	Strategic research (V)	Applications (VI)
	1. COREA, LASSY, SoNaR	1. DAESO, Daisy, DuOMAn	1. Daisy, DuOMAn
	2. IRME, Cornetto, DuOMAn	2. Daisy, DuOMAn, PACO-MT,	2.
	3. DPC	3.	3. DAESO, Daisy
		4. PACO-MT	4. PACO-MT
			5.

of semantic relationships (at the sentence level) and sentence fusion (priority V.1). These are useful for QA applications, information extraction and summarisation (priorities VI.1 and VI.3). These latter two topics, albeit on the discourse level, were also addressed by the Daisy project. DuOMAn produced (web) opinion mining tools (priority VI.1). DPC built a trilingual parallel corpus (priority IV.3) that can be useful for machine translation systems, such as Paco-MT (priority VI.4). Many corpus projects used the Alpino parser to produce syntactic annotations. Hence, even if no single project aimed at re-implementing a robust parser, as the SPRAAK project did for a robust speech recogniser, the Alpino robust syntactic parser has been improved and extended in several ways by various STEVIN projects (priority V.4).

Still, not all the priorities could be addressed: e.g., the lack of a tool for morphological analysis for derivation and compounding (priority V.3) and the absence of a text-based educational system (priority VI.5) are considered as lacunas. Also, more projects related to the semantic web (priority VI.2) would have been welcome, even if Cornetto, which created a lexical semantic database, is surely of high relevance for semantic web applications in Dutch. The BLARK for Dutch report [20] also listed the creation of benchmarks as an important action line (cf. Chap. 2, Sect. 2.2.3, p. 24). The STEVIN work programme did not retain this

topic as a priority item. However, the work programme text did state that projects could receive funding for the creation of test suites, test data and benchmarks if used for project internal systematic evaluation. Some of these data sets and test suites are available and can serve as reference data for other researchers. Only one specific project dedicated to creating a benchmark, c.q. for speech recognisers, was proposed (and awarded): NBest – cf. Chap.15, p. 271.

Some of the STEVIN priorities have been achieved by other research programmes, amongst others, the IMIX programme (Interactive Multimodal Information eXtraction.[5]) The IMIX programme, solely financed by the Netherlands (NWO), focussed on multimodal dialogue management for a medical QA system [25] and a non domain specific QA system called Joost [4] (priority III.3). IMIX started in 2002 while STEVIN was still under preparation. Therefore, it was later on agreed amongst the funding organisations that STEVIN projects could benefit from IMIX results, and that STEVIN would not explicitly target the IMIX topics. Funding agencies continued to award national projects that dealt with monolingual multimodal information extraction: AMASS++ (IWT-SBO)[6] (priorities V.1, VI.1, VI.1.3 and III.1), searching in radio and television multimedia archives: BATS (IBBT & ICT-Regie Im-Pact)[7] (priorities I.2 and III.1–2), compiling the CHOREC corpus of speech of children (IWT-SBO)[8] (priorities I.4.1, I.4.3, II.1, II.4 and VI.5), semantic web applications such as Kyoto[9] (EC-FP7) (priority VI.2) etc.

Thus, all in all, the STEVIN priorities are to a very large extent achieved. Concerning the creation of a digital language infrastructure, STEVIN is even cited as "probably the best example of a BLARK initiative for a tier 2 languague" [5, p. 1805]. Nevertheless, the topics not addressed during STEVIN have to be retained as themes for subsequent R&D funding initiatives, or at least their priority status is to be reconfirmed.

1.3.2 Improving the Scientific Capacity

Not only the coverage of each scientific priority in isolation constitutes a success indicator for STEVIN, but also the degree of "convergence" between the project results highly matters. For example, it would not be sensible to improve a syntactic parser for annotation purposes if the annotation schema used (strongly) differs from annotation schemas used by corpus projects. Also, technological components have to be (backwards) compatible and easily integratable with other lingware modules or larger frameworks. It is quite irrelevant and useless to implement a proper

[5] http://www.nwo.nl/nwohome.nsf/pages/NWOP_5ZLCE8_Eng

[6] http://www.cs.kuleuven.be/groups/liir/projects/amass/

[7] http://www.ibbt.be/en/projects/overview-projects/p/detail/bats

[8] http://www.esat.kuleuven.be/psi/spraak/projects/SPACE/

[9] http://cordis.europa.eu/fp7/ict/content-knowledge/projects-kyoto_en.html

noun speech synthesis module that can only function in a standalone way. In such a case, that particular scientific STEVIN priority could have been well covered, but the value for the overall HLTD community (academia and industry) might be fairly limited. The build-up of scientific capacity, including a digital language infrastructure, is not about re-inventing the wheel but rather about "standing on the shoulders of giants".

Figure 1.1 shows all the STEVIN scientific projects (cf. Sect. 1.2.1), four earlier (important) HLTD resources (projects A–D),[10] and four of the demonstrators (cf. Sect. 1.2.2 – DiaDemo (i), AAP (ii), NeOn (iii) and Hatci (iv)) that integrate STEVIN scientific results. The figure shows that STEVIN scientific projects do not constitute "islands", but that the results are shared, re-used and improved by other scientific projects and even – if the time lines permitted – integrated into end-user applications.

Figure 1.1 shows that more *resources* (the BLARK for Dutch layer) for speech have been created prior to STEVIN. The CGN, the Spoken Dutch Corpus developed earlier (project A [15]),[11] is the current reference corpus for spoken Dutch. Therefore, efforts on speech resources could be limited to extending the CGN corpus for specific target groups (JASMIN-CGN – project 4). The HMM speech recogniser (project B by the KU Leuven) has been upgraded into the SPRAAK package (project 2). The open source software of Praat (project C by the University of Amsterdam) has been extended in the STEVINcanPRAAT project (project 3).

Regarding textual resources, some catching-up had to be done. Hence, quite some STEVIN projects have created annotated textual corpora and lexica. The many connections between all the STEVIN corpus projects (cf. Fig. 1.1) show a high degree of interrelatedness. In particular, SoNaR (project 11) with its pilot project (project 5), is meant to become *the* reference written language corpus for Dutch. All these corpus efforts additionally resulted in extensive expertise in what is usually considered to be "trivial" issues such as data acquisition, IPR clearing and licence handling. These issues are in fact far from trivial (cf. [19]). On the contrary, the subsequent exploitation and dissemination of a corpus crucially depend on it. This kind of knowledge surely can be considered as a valuable resource for a digital language infrastructure – albeit of a different nature. The Alpino syntactic parser (project D by the University of Groningen) open source package has been used, adapted and extended by several STEVIN projects, mainly LASSY (project 6) and PACO-MT (project 16).

The pre-STEVIN materials already established themselves as the reference resource or tool (of their kind) in the Low Countries. Also their extensions (JASMIN-CGN, STEVINcanPRAAT and the various Alpino adaptations) will most probably "inherit" the success of the ancestor. And Fig. 1.1 clearly illustrates the importance of the SPRAAK tool kit for the field.

[10] Pre-STEVIN projects are shown in grey.

[11] The CGN is owned by the Dutch Language Union and maintained and made available by the HLT Agency – cf. Chap. 21.

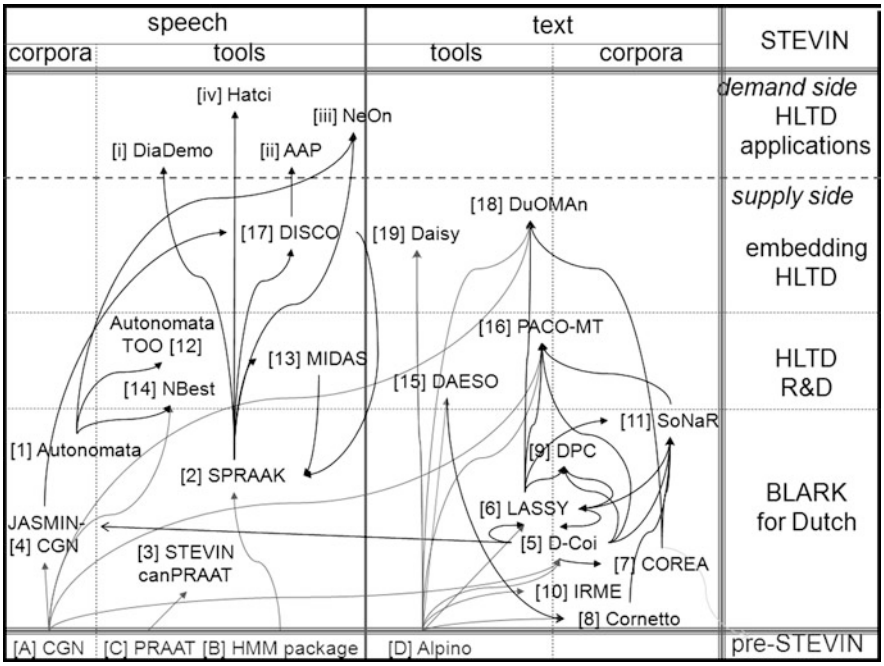


Fig. 1.1 A dependency graph showing how (pre-)STEVIN (scientific) projects are interrelated – projects are classified according to their most important results

The *HLTD R&D* layer presents a different situation for speech vs. text. In the speech processing area, several commercial TTS-engines (offering Dutch) exist (albeit as proprietary systems and "closed" source). The focus was put on improving the robustness of a speech recogniser and the treatment of proper nouns. The additional modules for proper noun pronunciation implemented by Autonomata (project 1) and Autonomata Too (project 12) can be used on top of a major standard commercial TTS package. Components of MIDAS have been integrated into SPRAAK to enhance the robustness to noise of the speech recognition tool kit. In the text domain, parsers and tagger/lemmatisers already exist in greater number. The research focus for STEVIN was thus placed on areas such as hybrid machine translation (PACO-MT – project 16), sentence fusion and detection of semantic overlap (DAESO – project 15).

STEVIN's *HLT embedded* text projects and applications (DuOMAn – project 18 and Daisy – project 19) were building to a lesser extent on previously developed STEVIN basic resources than is the case for the speech domain (DISCO – project 17) due to timing conflicts, even if some re-usage of materials did occur. Also, both in Flanders and the Netherlands, more research groups are working on text technology, all having their own tools based on different methods and principles (e.g., hand crafted rules vs. rules generated by machine learning techniques). In many cases, these have been adapted to Dutch so that the variety of tools available

is higher. Less de facto standard software packages exist in this domain – the Alpino parser being a notable exception.

However, it is to expected that in the future more tools and standards will establish themselves as de facto reference material. By the intermediary of CLARIN-NL,[12] standard file formats will most probably become widely used by the HLTD community, which will enhance the exchangeability of data between the various tools. Actually, the CLARIN-VL-NL[13] project with the name TTNWW, jointly funded by Flanders and the Netherlands, precisely aims at establishing standard formats to ensure the interoperability of tools during the execution of HLTD work flow processes. Many STEVIN materials are re-used in various CLARIN-NL projects. Hence, it is valid to state that STEVIN materials have effectively and substantially contributed to the build-up of HLTD capacity in the Low Countries. And it is equally safe to expect that STEVIN materials will remain important for the field in the near future. We refer the reader to the overall concluding chapter (Chap. 22, p. 395) for more reflections on the international context of STEVIN and for an outlook for future HLTD activities.

1.4 Organisation of This Volume

The remainder of this volume is organised as follows. A more detailed account from a policy point of view on the STEVIN programme is offered in the second chapter of this volume (Part I). The chapters on the STEVIN scientific projects are grouped into three parts in line with Table 1.3: resource related (II), technology or research related (III) and application related (IV). In a separate chapter, the HLT Agency, which is responsible for the IPR management, the maintenance and distribution of the STEVIN results, presents itself. Together with a concluding and forward looking chapter, it constitutes Part V.

References

1. Akkermans, J., van Berkel, B., Frowein, C., van Groos, L., Van Compernolle, D.: Technologieverkenning nederlandse taal- en spraaktechnologie. Techincal report., M&I/Partners & Montemore, Amersfoort & Leuven (2004) (in Dutch)
2. Arnold, E., Kuhlman, S.: Rcn in the norwegian research and innovation system. In: Background report no. 12 in the Evaluation of the Research Council of Norway, Fraunhofer ISI, Karlsruhe (2001)

[12] http://www.clarin.nl

[13] http://www.ccl.kuleuven.be/CLARIN/pilot.htm – in Dutch

3. Boersma, P.: PRAAT, a system for doing phonetics by computer. Glot Int. **5:9/10**, 341–345 (2001)
4. Bouma, G., Muri, J., van Noord, G., van der Plas, L.: Question-answering for dutch using dependency relations. In: Proceedings of the CLEF 2005 Workshop, Vienna (2005)
5. Boves, L., Carlson, R., Hinrichs, E., House, D., Krauwer, S., Lemnitzer, L., Vainio, M., Wittenburg, P.: Resources for speech research: present and future infrastructure needs. In: Proceedings of Interspeech 2009, Brighton (2009)
6. Cucchiarini, C., Driesen, J., Van hamme, H., Sanders, E.: Recording speech of children, non-natives and elderly people for HLT applications: the JASMIN-CGN corpus. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Marrakech (2008)
7. De Belder, J., Moens, M.F.: Integer linear programming for Dutch sentence compression. In: Gelbukh A. (ed.) Proceedings of CICLing 2010, Lasi. Lecture Notes in Computer Science, pp. 711–723. Springer, Berlin, Heidelberg (2010)
8. Demuynck, K., Roelens, J., Van Compernolle, D., Wambacq, P.: SPRAAK: an open source speech recognition and automatic annotation kit. In: Proceedings of the International Conference on Spoken Language Processing, Jeju, Korea (2008)
9. Gemmeke, J., Van hamme, H., Cranen, B., Boves, L.: Compressive sensing for missing data imputation in noise robust speech recognition. IEEE J. Sel. Top. Signal Process. **4**(2), 272–287 (2010)
10. Grégoire, N.: Duelme: a Dutch electronic lexicon of multiword expressions. J. Lang. Resour. Eval. Special issue on Multiword Expressions. **44**(1-2), 23–40. Springer (2010)
11. Hendrickx, I., Bouma, G., Coppens, F., Daelemans, W., Hoste, V., Kloosterman, G., Mineur, A.M., Van Der Vloet, J., Verschelde, J.L.: A coreference corpus and resolution system for Dutch. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Marrakech (2008)
12. Kessens, J., van Leeuwen, D.: N-Best: the northern and southern Dutch evaluation of speech recognition technology. In: Proceedings of Interspeech 2007, Antwerp, pp. 1354–1357 (2007)
13. Krauwer, S.: The Basic Language Resource Kit (BLARK) as the first milestone for the Language Resources Roadmap. In: Proceedings of the International Workshop Speech and Computer 2003, Moscow (2003)
14. Marsi, E., Krahmer, E.: Detecting semantic overlap: a parallel monolingual treebank for Dutch. In: Proceedings of Computational Linguistics in the Netherlands (CLIN 2007), Nijmegen (2007)
15. Oostdijk, N.: The design of the Spoken Dutch Corpus. In: Peters P., Collins, P., Smith, A. (eds.) New Frontiers of Corpus Research. Rodopi, Amsterdam/New York (2002), pp. 105–112
16. Oostdijk, N., Reynaert, M., Monachesi, P., van Noord, G., Ordelman, R., Schuurman, I., Vandeghinste, V.: From D-Coi to SoNaR: a reference corpus for Dutch. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Marrakech (2008)
17. Paulussen, H., Macken, L., Truskina, J., Desmet, P., Vandeweghe, W.: Dutch Parallel Corpus a multifunctional and multilingual corpus. Cahiers de l'Institut de Linguistique de Louvain **32.1-4**, 269–285 (2006)
18. Réveil, B., Martens, J.P., van den Heuvel, H.: Improving proper name recognition by adding automatically learned pronunciation variants to the lexicon. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010), Valletta (2010)
19. Reynaert, M., Oostdijk, N., De Clercq, O., van den Heuvel, H., de Jong, F.: Balancing SoNaR: IPR versus processing issues in a 500-million-word written Dutch Reference Corpus. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010), Valletta (2010)
20. Strik, H., Daelemans, W., Binnenpoorte, D., Sturm, J., de Vriend, F., Cucchiarini, C.: Dutch HLT resources: from BLARK to priority lists. In: Proceedings of ICSLP, Denver (2002)

21. Strik, H., Cornillie, F., Colpaert, J., van Doremalen, J., Cucchiarini, C.: Developing a CALL system for practicing oral proficiency: how to design for speech technology, pedagogy and learners. In: Proceedings of the SLaTE-2009 Workshop, Warwickshire (2009)
22. Tsagkias, E., Weerkamp, W., de Rijke, M.: News comments: exploring, modeling, and online predicting. In: Proceedings of the 2nd European Conference on Information Retrieval (ECIR 2010), pp. 109–203. Springer, Milton Keynes, UK (2010)
23. van Noord, G.: Huge parsed corpora in LASSY. In: Van Eynde, F., Frank, A., De Smedt, K., van Noord G. (eds.) Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7), Groningen (2009). LOT Occasional Series
24. van Noord, G.: Learning efficient parsing. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009), Athens (2009). Association for Computational Linguistics
25. van den Bosch, A., Bouma, G. (eds.): Interactive Mulit-modal Question-Answering. Theory and Applications of Natural Language Processing. Springer, Heidelberg/New York (2011)
26. van den Heuvel, H., Martens, J.P., D'Hoore, B., D'Hanens, K., Konings, N.: The Autonomata spoken name corpus. design, recording, transcription and distribution of the corpus. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Marrakech (2008)
27. van den Heuvel, H., Martens, J.P., Konings, N.: Fast and easy development of pronunciation lexicons for names. In: Proceedings of LangTech, Rome (2008)
28. van Leeuwen, D., Kessens, J., Sanders, E., van den Heuvel, H.: Results of the N-Best 2008 Dutch speech recognition evaluation. In: Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech 2009), Brighton. International Speech Communication Association, pp. 2571–2574 (2009)
29. Vandeghinste, V.: Scaling up a hybrid mt system: from low to full resources. Linguist. Antverp. **7**, 65–80 (2008)
30. Vossen, P., Agirre, E., Calzolari, N., Fellbaum, C., Hsieh, S., Huang, C., Isahara, H., Kanzak, K., Marchetti, A., Monachini, M., Neri, F., Raffaelli, R., Rigau, G., Tescon, M.: Kyoto: a system for mining, structuring and distributing knowledge across languages and cultures. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), Marrakech (2008)
31. Vossen, P., Maks, I., Segers, R., van der Vliet, H., van Zutphen, H.: The Cornetto Database: the architecture and alignment issues of combining lexical units, synsets and an ontology. In: Proceedings of the Fourth International GlobalWordNet Conference, Szeged (2008)

Part I
How It Started

Chapter 2
The STEVIN Programme: Result of 5 Years Cross-border HLT for Dutch Policy Preparation

Peter Spyns and Elisabeth D'Halleweyn

2.1 Context

Dutch is ranked as the 40th most widely spoken language of the world's 6,000 languages. Most of the 23 million Dutch native speakers live in 2 neighbouring territories, the Netherlands and the Flemish region of Belgium, that have Dutch as an official language. As language policy decisions taken on one side of the national border affect citizens at the other side of the border, the Netherlands and Belgium created in 1980 the Dutch Language Union (Nederlandse Taalunie – NTU) as an intergovernmental organisation to conduct a joint language policy. The NTU's mission is to deal with issues concerning the position of the Dutch language. It includes Dutch language and literature as a scientific subject, Dutch as a cultural language, Dutch as an administrative language, Dutch as a means of communication, and, more in general, the Dutch language as a tool for social interaction.

In an ICT based society a language needs a digital infrastructure (digital corpora and dictionaries, software and lingware modules, etc.) to maintain its position as a "used and useful" language and to avoid what is called an "electronic Gutenberg" effect. However, the market for human language technology for Dutch (HLTD) is seemingly too limited to attract important investments by industry in HLTD. As a consequence, the Flemish and Dutch governments decided in 2004 to spend

P. Spyns (✉)
Nederlandse Taalunie, Lange Voorhout 19, 2514 EB Den Haag, Nederland
e-mail: pspyns@taalunie.org

Vlaamse overheid – Departement Economie, Wetenschap en Innovatie, Koning Albert II-laan 35, bus 10, B-1030 Brussel, België
e-mail: Peter.Spyns@ewi.vlaanderen.be

E. D'Halleweyn
Nederlandse Taalunie, Lange Voorhout 19, 2514 EB Den Haag, Nederland
e-mail: edhalleweyn@taalunie.org

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_2, © The Author(s) 2013

11.4 million euros to stimulate the HLTD sector (industry and academia) and thus strengthen the position of Dutch in the modern knowledge based society [11].

2.2 Historical Background

2.2.1 Researching Apart

HLT for Dutch started early. It began to flourish in the 1980s thanks to the Eurotra programme of the EC. Eurotra was intended as an EC research and technology development effort targeted at the development of a machine translation system for its internal use. For Dutch, it was mainly a collaboration between the universities of Leuven and Utrecht [18]. In parallel, some private companies also funded research on machine translation.[1] This initial wave slowly faded away in the beginning of the 1990s. In the Netherlands, the Dutch Organisation for Scientific Research (NWO) initiated some large scale programmes on HLT for Dutch.[2] In Flanders, no specific HLT research programmes were set up, except for the Flemish Research Initiative in Speech and Language Technology[3] that ran from 1994 till 1997. In Flanders research funding is mainly organised in a "bottom up" manner, not thematically or programmatically as is mainly the case in the Netherlands.

Of course, researchers in both Flanders and the Netherlands collaborated in cross-border projects, but this happened on a personal and ad hoc basis. In addition, researchers were not always aware of the availability of resources and tools for Dutch developed elsewhere. Systematically sharing and maintaining of resources hardly occurred. How the Eurotra software, which represented a research effort of more than a decade, fell into oblivion is an all too sad example. Clearly a coordinating platform or organisation was lacking.

2.2.2 Researching Apart Together

Things changed in the 1990s. The Flemish and Dutch governments became interested in HLTD at the start of the 1990s. They initiated research programmes, albeit still separate, and organised some exploratory policy studies. For example, in

[1] Philips Eindhoven: the Rosetta system; Bureau voor SysteemOntwikkeling (BSO): the Distributed Language Translation system (DLT); and Siemens: the METAL system.

[2] SPIN (1984–1995), CELEX (1986–2000), and the HLT priority programme (1995–2000).

[3] www.vrwi.be/pdf/advies38.pdf

a large technology foresight exercise of 1998,[4] HLT was mentioned as a potentially strategic technology domain for the economy of the Netherlands.

In the mid 1990s, the EC sponsored the Euromap Language Technologies project (1996–2003). Euromap aimed at accelerating awareness of the benefits of HLT enabled systems, services and applications within user sectors, policy makers and national administrations and bridge-building and market-enabling services to stimulate market take-up of HLT RTD projects' results. The Euromap project wanted to determine the status of HLT for the various languages in the participating countries. For each participating territory, a national profile was made as well as a national policy review [13]. In addition, directories of all research groups and companies active in the field of HLT of a country were published – e.g., cf. [10] for the Flemish directory. Initially, the Flemish and Dutch administrations participated in Euromap as separate partners.

The objectives of Euromap ran partly in parallel with the ambitions of the NTU that prepared and published in 1998 a study on the status of Dutch in speech and language technology[6].[5] The fact that the NTU became the "national" focal point for the entire Dutch language, representing both Flanders and the Netherlands in the second phase of the Euromap project, gave a boost to the implementation of some of the recommendations of this study. The national seminars on various HLT related subjects organised in the framework of Euromap, for example largely contributed to network building and laid the foundation for the future cooperation between academia, industry and governments in the Netherlands and Flanders.

In addition, as these were the booming years of Lernout&Hauspie Speech Products in Flanders,[6] HLT became very prominent on the public forum in the Low Countries. The study and these (economic) circumstances made the NTU – and the Dutch and Flemish governments – realise the importance of a digital language infrastructure for the Dutch language. At that time such an infrastructure was largely lacking. As a result an HLT for Dutch Platform (HLT Platform) in which the relevant government departments and agencies were represented, was installed in 1999 [2]. The goals of the HLT Platform, which constituted a forum for information exchange, agenda adjusting and joint activities, were:

- To promote the position of the Dutch language in HLT developments, so that the Dutch language could become and remain a "first class citizen" language within a multilingual European information society;
- To establish the proper conditions for a successful management and maintenance of basic HLT resources developed with governmental funding;

[4] See http://www.rand.org/pubs/rand_europe/RE98004.1

[5] A summary in English can be found in [7].

[6] L&H became the top worldwide player in HLT before collapsing due to financial fraud and mismanagement. Nuance International Communications can be considered as its "partial successor".

- To promote and stimulate the collaboration between the research community and the business community in the field of HLT;
- To contribute to European collaboration in HLT-relevant areas;
- To establish a network, both electronic and personal, that brings together demand and supply of knowledge, products and services.

In parallel, the NTU took on the challenge to coordinate the development of high quality resources needed for automated translation from and into Dutch for the Systran translation system. This was the TransLex project [12], funded by the EU MLIS programme with additional contributions by the Flemish and Dutch governments together with the private partner Systran and the translation service of the EC.

In 1998, the construction of a Spoken Corpus for Dutch (CGN)[17] started. Again, Flemish and Dutch governments have jointly financed the project. The NTU received the ownership of the corpus and became responsible for its maintenance and exploitation. However, the NTU did not assume any central role in the process. Note that, from the governance point of view, only the CGN board (of funding organisations) and the scientific steering group were organised as a joint endeavour. All other (practical) matters (set-up, funding etc.) were organised separately in Flanders and the Netherlands. The CGN scientific steering group ensured that scientific activities remained compatible (common formats, protocols, tools etc.).

2.2.3 Researching and Developing Together

The NTU published in 1999 together with the HLT Platform an "action plan for Dutch in speech and language technology". Four major action lines were defined:

- Action line A: setting up an information brokering service;
- Action line B: strengthening the digital language infrastructure;
- Action line C: defining standards and evaluation criteria;
- Action line D: developing a management, maintenance and distribution plan.

Several working groups, consisting of researchers from academia and industry, started to write specific plans on how to accomplish these four action lines. *Action line A* has been taken up by the NTU and resulted in the creation of the HLT Info desk.[7] The HLT Info desk publishes a newsletter, maintains a website with an overview of HLTD related organisations (academia, industry and government) and HLTD events in Flanders and the Netherlands.

Action line B has eventually materialised in an HLTD R&D programme. Extensive preparatory activities paved the way for this programme. Field surveys resulted in the description of a basic language resource kit (BLARK) for Dutch. A BLARK is

[7] See http://taalunieversum.org/taal/technologie/ – in Dutch.

defined as the set of basic HLT resources that should be available for both academia and industry [15]. Not only were all the materials (data, modules and tools) available (or at least identified) at that moment listed, but also "missing links" ' were identified and included in the overview. Prioritisation exercises, including discussions and meetings with the entire HLTD field, led to ranked lists of R&D topics [4, 8, 23]. A longer term road map was sketched [3].

In addition, the Dutch Ministry of Economic Affairs has ordered a specific HL technology forecast to estimate the economic value and potential of HLTD and to determine the ideal government intervention logic [1]. Eventually, a proposal for a joint Flemish-Dutch R&D programme was drafted. The proposal was baptised STEVIN (Essential Resources for Speech and Language Technology for Dutch). The proposal was accepted and STEVIN started in September 2004.

Activities on *action line C* have been combined with action line B: determining whether materials are available could not be done without a quality evaluation. However, actual evaluation criteria or benchmarks have not been developed – except as (parts of) projects in the STEVIN-programme (e.g., the NBest project [14] – cf. Chap. 15, p. 271).

The working group for *action line D* has delivered a blueprint for management, maintenance, and distribution of publicly funded HLT resources that eventually resulted in the creation of the HLT Agency for Dutch by the NTU [3]. ELDA and LDC served as examples. This agency acts as a "one-stop-shop for HLTD" and takes care of maintaining, distributing and promoting HLT for Dutch project results (corpora, tools, dictionaries etc.) [25] – cf. Chap. 21, p. 381.

2.3 The STEVIN Programme

2.3.1 In a Nutshell

In line with the action plan of the HLT Platform (cf. Sect. 2.2.3), the STEVIN-programme aimed to contribute to the progress of human language technology for Dutch (HLTD) in Flanders and the Netherlands and to stimulate innovation in this sector. In addition, it aimed to strengthen the economic and cultural position of the Dutch language in the modern ICT-based society. The mission of the programme was translated into three specific main goals:

1. Build an effective digital language infrastructure for Dutch, based on the BLARK priorities for Dutch;
2. Carry out strategic research in the field of language and speech technology, especially in areas of high demand for specific applications and technologies;
3. Advance the creation of networks and the consolidation of language and speech technology activities, educate new experts, stimulate the demand for HLT products.

The STEVIN HLT programme was comprehensive in many respects. First of all, because it was based on co-operation between government, academia and industry both in Flanders and the Netherlands. For example, projects with partners from both Flanders and the Netherlands were encouraged. Co-operation saves money and effort by avoiding duplication of activities and enhances scientific excellence thanks to an increased competition. Secondly, the programme encompassed the whole range from basic resources to applications for language users. For example, application oriented projects were encouraged to build upon results of the resource oriented projects (cf. Chap. 1, Sect. 1.3.2 p. 12). And thirdly, it paid attention to the distribution, dissemination and valorisation of project results by means of the HLT Agency (cf. Chap. 21, p. 381). To ease the distribution of the resulting resources and tools, the HLT Platform stipulated the *obligation* to transfer the ownership of the foreground results (i.e. material made in the course of a STEVIN project) to the NTU, which is rather uncommon. The important task of clearing the IPR and issuing licence agreements for further use was delegated to the HLT Agency, which in turn received some extra funding from the STEVIN budget.

2.3.2 Governance

Various committees were set up around the STEVIN programme, as Fig. 2.1 shows. The HLT Platform (cf. Sect. 2.2.2) became the *HLT board* supervising STEVIN. It consisted of the NTU and the funding bodies.[8] The NTU was the overall coordinating instance.

A *programme committee – PC*, consisting of both academic and industrial representatives, was responsible for all scientific and content related issues. It consisted of local Flemish and Dutch HLT experts who wrote a detailed multi-annual research programme (topics, expected outcomes, instruments, timing, ...). The PC defined the various calls for project proposals. An *international assessment panel* (IAP) of eight highly respected HLT-experts evaluated the submitted R&D project proposals. The PC added a "local check" to the assessment of the IAP. Divergences of opinion between the IAP and the PC were rare and of minor importance. Several calls (three open calls and two calls for tender) have been issued over time.

Next to the R&D projects, which were supposed to achieve the first two main goals of STEVIN mentioned above, some other (smaller sized) actions were initiated by the *"accompanying activities" working group*:

[8]The HLT Platform members were, next to the NTU, the Flemish Department of Economy, Science and Innovation (EWI), the Flemish Agency for Innovation by Science and Technology (IWT), the Fund for Scientific Research – Flanders (FWO), the Dutch Ministry of Education, Culture and Sciences (OCW), Innovation NL (the Dutch innovation agency) representing the Dutch Ministry of Economy, Agriculture and Innovation (ELI)), and the Netherlands Organisation for Scientific Research (NWO).

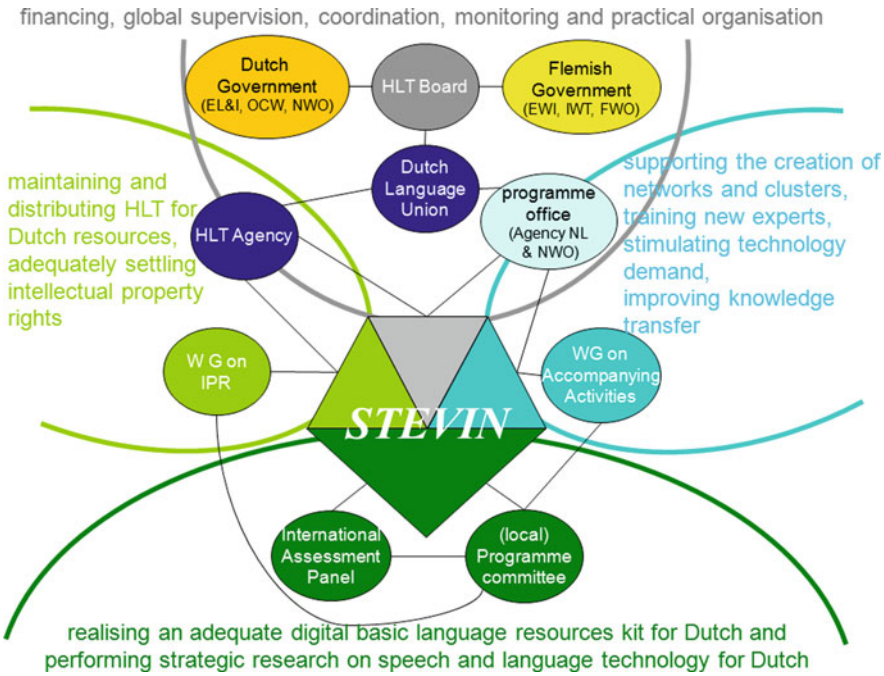


Fig. 2.1 Main goals of STEVIN and distributed responsibilities

- Demonstration projects had to increase the demand for HLT technology by using proven HLT technologies to build highly visible close-to-market applications;
- Educational projects aimed at sensitising young students within educational settings (school, museums, etc.) for the possibilities of language and speech technologies;
- Master class projects targeted high level decision makers within government organisations and the industry to familiarise them with the opportunities offered by HLT.

STEVIN has awarded 19 R&D projects (in total 8.909 K euros), 14 demonstrator projects (1.011 K euros), 3 educational projects (100 K euros), 2 master classes (33 K euros) and 31 networking grants (45 K euros in total). The acceptance rate for the R&D projects was between 26 and 33 %. In the Low Countries, most of the funding agencies consider an acceptance rate of around 30 % sufficiently selective to guarantee scientific excellence and high enough to fund (almost) all the best proposals.

A *programme office*, a joint collaboration of the Netherlands Organisation for Scientific Research and the Dutch innovation agency called Agency NL, took care of the operational matters, such as the practical organisation of the calls (submission site, related documents etc.)

An important committee was the *IPR working group* that defined the licence templates. These licences settled the ownership transfer of the foreground results to the NTU, the conditions under which third parties agreed to make their resources available for academia (and if possible also for industry), the permissions for STEVIN researchers to continue to work on their material and the terms of usage for third parties to use the STEVIN resources and tools. As a result, the IPR of all the STEVIN material has been legally cleared, which opened a wide range of different possibilities to distribute and exploit the material. This task, managing the IPR of the STEVIN results, was delegated by the NTU to the *HLT Agency* [25] – cf. Chap.21, p. 381.

2.3.3 Monitoring and Evaluation

Some time after the start of the STEVIN programme a *baseline* was defined [1]. It encompassed the then current state of HLT for Dutch in terms of number of researchers active, turn-over of HLT companies, degree of academia-industry cooperation, degree of Flemish-Dutch cooperation etc. in the spirit of the Euromap country reports ([13] – cf. Sect. 2.2.2). This base line served as reference point for the final evaluation to determine to which extent the STEVIN programme had a positive impact on HLTD in Flanders and the Netherlands. During the programme, a light weight *monitoring* process at project level was organised. Each project had to organise two site visits during which two members of the PC attended presentations on the project's progress and achievements. The members of the PC studied the reports, gave suggestions and made critical remarks – if needed. Additionally, the projects, if appropriate, had to organise an external validation exercise or deliver some "circumstantial evidence" of a quality control check (e.g., a test report by a research group not belonging to the consortium that had used the resource concerned).

Half way through the programme, a scientific *mid term evaluation* by the IAP was organised to see if the entire programme was on track and if any adjustments had to be made [22]. In addition, the PC made a self evaluation report. A.o. the IAP felt that STEVIN material was worthy of more high profile scientific publications (the projects and the programme in its entirety) than was the case at that moment. Another matter of concern was the lack of projects in the multimodal and/or multimedia and semantic domains. But all in all, the IAP in its report[9] congratulated the HLTD community in the Low Lands on their achievements within the STEVIN programme [19].

The *final evaluation* was concluded before the actual end of the programme. As a consequence, some projects still produced an important number of publications that were not taken into account. An important advantage would have been that a

[9] Available in English via www.stevin-tst.org/english.

smooth continuation had been enabled as the funding authorities already had almost all the necessary information available to decide on follow-up activities before the actual end of the programme. Unfortunately, the aftermath of the 2008 financial crisis decided otherwise.

2.3.4 The Final Evaluation

The final evaluation of the STEVIN programme was an overall evaluation. Not only the scientific issues but also the governance and economic aspects of the programme were taken into account. A small ad hoc committee did the preparatory work, largely inspired by the evaluation framework and practices of the Flemish Department of Economy, Science and Innovation.

2.3.4.1 Evaluation Assignment

The HLT board formulated a set of evaluation questions, which can be grouped into four major categories.

- Efficiency: Were the resources properly and adequately used? Was the management of the programme efficient? and Was the programme adequately monitored?
- Effectiveness: Did the programme achieve its targets? Was the programme effectively organised? Did the programme influence the policy agenda in Flanders and The Netherlands?
- Usefulness: Were the problems in the HLT domain identified at the start of the programme successfully addressed? Was there an overlap with other activities/-efforts? and Which role did STEVIN play in the HLT field, both nationally and internationally?
- Relevance: To what extent did STEVIN lead to usable material for the HLT field and user groups? To what extent technological and scientific progress in the HLT field did evolve thanks to STEVIN ? and What was the added value of STEVIN?

These evaluation questions were grouped around the major issues at play in the STEVIN-programme:

- Governance and management of the programme;
- Application and selection process of project proposals;
- Effects and impacts of the programme;
- Positioning of the STEVIN programme with respect to other programmes;
- Future of the programme.

In order to obtain an objective evaluation, a call for tender was issued and an external consultant (c.q. the Technopolis group[10]) [9] was selected to perform the evaluation. The same questions were addressed by the PC as well in their self assessment report.[11]

2.3.4.2 Evaluation Methodology

A combination of both quantitative and qualitative research methods was employed [9]. During a desk research phase, the consultant analysed all relevant documents (the STEVIN multi-annual work plan, yearly reports, the baseline and mid-term reports, self assessments reports, fact files, meeting minutes of the various committees, call for proposal texts, etc.). The STEVIN programme was also compared with some other (foreign) programmes – in particular concerning the governance structure, the financial management and selection, evaluation and monitoring procedures. An international expert panel (other than the STEVIN IAP mentioned earlier) assessed the scientific output of the programme. In total, 127 relevant contacts were invited to participate in two online surveys (cf. Sect. 2.3.4.3). A network analysis was used to map the various co-operation relationships within the STEVIN-programme. Finally, 23 interviews were held that involved the most important STEVIN stakeholders. More details can be found in [21].

2.3.4.3 Evaluation Outcomes

Of course it is impossible to report here[12] on all aspects of the STEVIN final evaluation. We limit ourselves to summarising the assessment of the three main goals of the programme (cf. Sect. 2.3.1), as well as giving some general comments of the international experts that concern the entire programme. In the subsequent Sect. 2.3.4.3, ten recommendations by the external evaluator are presented. We refer the reader to the chapter on the HLT Agency (cf. Chap. 21, p. 381) for more details on IPR and licence agreement management, and other issues on maintenance, distribution, promotion and utilisation of the STEVIN results. For a more detailed account on the scientific achievements, the reader is referred to the individual chapters on the various STEVIN projects in this volume. The introductory chapter (cf. Chap. 1, p. 1) gives a more global view on the entire programme, while overall conclusions and forward looking statements are provided in Chap. 22, p. 395.

[10] http://www.technopolis-group.com/site/

[11] See www.stevin-tst.org/programma#evaluaties: the complete evaluation report is in Dutch with a summary in English, but the PC self assessment report is in English.

[12] This section is largely based on the STEVIN final evaluation report by Technopolis group.

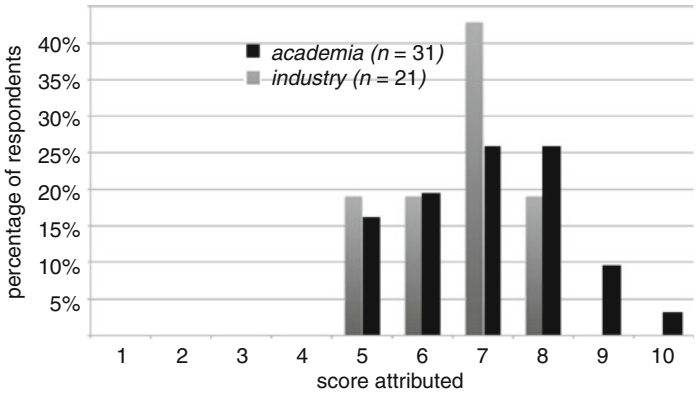


Fig. 2.2 Reported degree of achieved success for STEVIN main goal 1: setting up the BLARK for Dutch

Overall Assessment

Figures 2.2–2.4 summarise the scores of the two online surveys. The first survey addressed academics in Flanders and the Netherlands, c.q. successful and unsuccessful submitters of STEVIN proposals. Research institutes that had participated in the baseline survey, even without any link with STEVIN, were also invited. Sixty-two research institutes were contacted, of which 56.5 % responded. The second survey concerned the Flemish and Dutch HLT industry. Again, applicants for funding (granted or not) were invited as well as companies that had participated in the baseline survey. Sixty-five companies were contacted with a response rate of 43.2 %. The responses may thus safely be assumed to be representative (overall response rate of 49.6 %).

Applicants for STEVIN funding were asked to rate (on a scale of 1–10) the achievements and "mechanics" of the STEVIN programme – e.g., statements on the transparency of the decision process, the quality of communication, the expectations towards the programme etc. Participants in the baseline survey had to provide data and information on the status of HLT in their organisation – e.g., the number of HLT related staff, HLT turn-over (if applicable), expenditures in HLT R&D etc. A comparison between the situation described in the baseline report and the new situation should allow to assess the impact of the STEVIN programme on the domain. Due to space limitations, we cannot discuss the comparison in this volume.

Figure 2.2 shows that the participants of the STEVIN programme agreed that STEVIN largely succeeded in setting up a digital language infrastructure for Dutch, i.e. creating many of the missing building blocks of the BLARK for Dutch. The overall score is 6.6. Representatives of academia considered this mission as accomplished in a slightly higher degree than companies. And Flemish respondents are slightly more positive than respondents from the Netherlands.

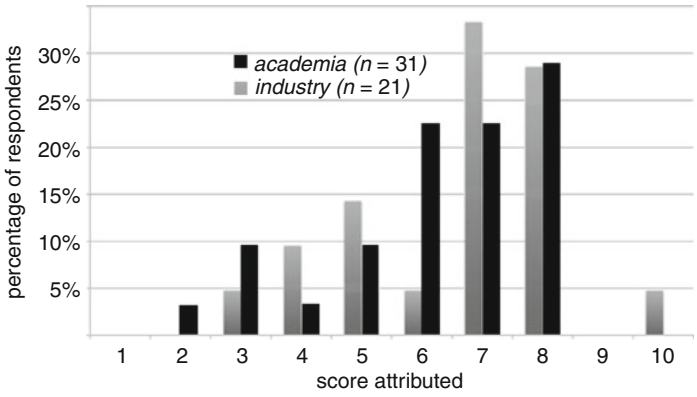


Fig. 2.3 Reported degree of achieved success for STEVIN main goal 2: performing strategic HLTD research

Figure 2.3 reflects how the participants assessed the opportunity offered by STEVIN to perform strategic basic research (= second main goal of STEVIN). Again, the overall score is 6.6. Academics are slightly less positive (in particular Flemish academics: 5.8). Again, the international experts involved in the evaluation pointed out (as did the IAP members during the mid-term review) that a programme as STEVIN should generate more international high profile publications. Nevertheless, they concluded that many STEVIN deliverables, even if not always cutting edge, were highly important to set up the BLARK for Dutch.

Even if a too low number of high profile scientific publications seems a justified point of critique, one has to take into account that creating elements of a digital language infrastructure does not necessarily imply performing cutting edge research – in some cases, it is rather the contrary. And in all fairness, it should be mentioned that around 55 new papers appeared[13] after the delivery of the evaluation report. In total, around 200 official STEVIN publications were published. Application oriented projects resulted in more higher impact publications than the resource producing projects.

Many of the corpora, tools, protocols, databases, etc. resulting from STEVIN still are, to an important degree, instrumental for the Flemish and Dutch local CLARIN counterparts of the European CLARIN project [26].[14] In addition, some of the STEVIN results are not only important for Dutch, but do also have an impact (practical and theoretical) on research on other languages. For example, the

[13]See www.stevin-tst.org/publicaties.php. In particular, the MIDAS (cf. Chap. 16) and DuOMAn (cf. Chap. 20) projects produced an impressive list of publications.

[14]The EU FP7 preparatory project Common Language Resources and Technology Infrastructure (CLARIN), in a nutshell, aims at facilitating e science, for the human and social sciences mainly by providing easy access to HL resources and giving support through HL tools.

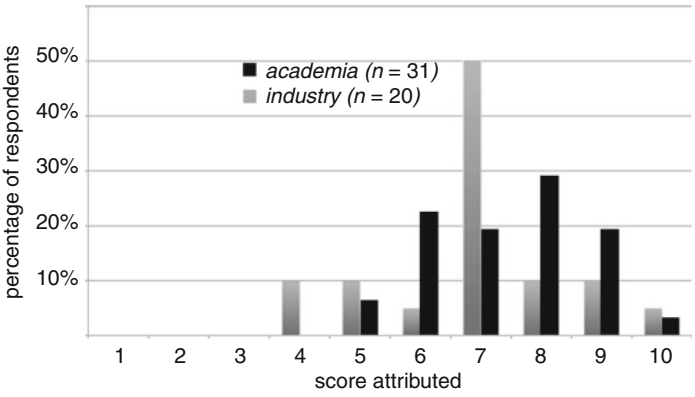


Fig. 2.4 Reported degree of achieved success for STEVIN main goal 3: HLTD networking and demand stimulation

STEVINcanPRAAT project (cf. Chap. 5, p. 79) improved the PRAAT tool [5] that is widely used by an international community.

Figure 2.4 shows that the stakeholders positively judged the impact of STEVIN on the strengthening of networks and co-operations in the Low Countries (= third main goal of the programme). Even if participants knew each other before their involvement in STEVIN, STEVIN offered the opportunity to actually co-operate. In addition, the intensity of co-operation between Flemish and Dutch parties and between academia and industry increased thanks to STEVIN. The fact that STEVIN created a unique opportunity for all HLT subdisciplines (speech, language, information extraction, dialogue, ...) was well appreciated. Flemish and Dutch respondents reacted alike. Respondents from academia (7.4) are more positive than respondents from the private sector (6.9). Jointly performing research and exchanging knowledge were the two most cited types of co-operation.

Overall, researchers apparently felt the need to set up the BLARK for Dutch more strongly than industry, and hence were more happy with the results. Companies were more interested in performing strategic research, acquiring and integrating new technology to improve their products, while researchers preferred to perform more basic research and publish papers instead. Usually academia is more open to co-operation than industry. These statements are confirmed by other findings (not mentioned here) in the survey.

The external evaluator compared STEVIN with a few other international R&D programmes.[15] The comparison showed that STEVIN is quite unique in its cross-border co-operation (including cross-border funding). STEVIN can be seen as a precursor of the "joint programming" approach of the EC [20]. Its comprehensive approach (ranging from strategic research to demonstrator projects and sensitisation

[15]The programmes were Nordite, IM-Pact, Npelt, ICT-Eprsc and Fit-IT, mostly ICT-related.

and promotional activities) and the involvement of various ministries and agencies at both sides of the border was rather rare. As such, the programme has achieved its objectives and is regarded by the evaluators as a successful programme.

Evaluation Recommendations

An evaluation of an R&D programme focusses not only on ex-post reflections (on how things have happened) but also includes ex-ante suggestions for post STEVIN activities. The external evaluator has condensed his findings in ten recommendations[16]:

1. The integrated approach of STEVIN was a good approach and should be replicated in a potential follow-up of STEVIN. The focus should then be shifted from the BLARK and strategic research to application-oriented research and demonstration projects. It is important to balance between the different types of research. In the design of the programme, multiple modalities should be possible: basic research combined with more application-oriented research and projects aimed at either strategic or application-oriented research. Maybe less of a priority, but still important are projects aimed at basic language infrastructure.
2. STEVIN is an example of transnational cooperation through "joint programming" that has value for both funders and performers. A possible follow-up to STEVIN should also have a bilateral structure with a "common pot".
3. The main structure of governance does not need to be adjusted. However, the tasks and responsibilities should be defined more precisely, so that it is clear to everyone what the tasks and roles of the various organisations involved are.
4. The programme office needs to be positioned more closely to the NTU. This could be done by means of a secondment to the NTU from various organisations.
5. The programme office should also be more balanced, in the sense that there is a better Dutch-Flanders balance in governance structure.
6. In general, partly dependent on the focus of a follow-up programme, the composition of different committees and commissions should be reviewed. If its focus is to be more on the application of HLT-knowledge in practice, representation of industry and applicators should be enforced.
7. IPR issues, including how to deal with open source, should be addressed before the start of a follow-up programme. Rules regarding IPR should be clearly defined and availability of standard contracts, etc. should also be taken into

[16]We copied and pasted the recommendations literally from the evaluation report to avoid any interpretation bias.

consideration. The preparations can build on the work of the IPR Working Group and the experiences of the HLT Agency.

8. A more active collaboration with related programmes at the national level, and at European level is needed in the follow-up programme. In addition, it is to be considered whether a junction is possible with social innovation programmes in the fields of education, care, and safety.
9. If strategic research obtains an important role in a follow-up programme, there should be a greater emphasis on publications in international journals and at international summits.
10. Consider dedicating part of the budget to an international publication in which the results of the STEVIN programme are presented in conjunction.[17]

2.4 Discussion

In this section, we briefly treat two governing principles that are seemingly very typical of the STEVIN programme. In fact, it is rather uncommon for researchers to have to transfer the ownership of their research results to a governmental organisation (cf. Sect. 2.4.2) and to be funded according to the actual delivery of results specified on beforehand instead of on the basis of a predefined amount of time and resources (cf. Sect. 2.4.1).

2.4.1 Delivering Results

Most of the contracts between a funding agency and research institutes are based on an obligation to perform to the best of one's abilities (= a commitment by a researcher to use as well as possible the given means to investigate a topic without any guarantee on success). STEVIN contracts however were based on an obligation to achieve results (= a commitment by a researcher to deliver well specified results). As the aim of STEVIN was to create a digital language infrastructure, the funding organisations did expect a finalised corpus or a properly working tool to be actually delivered at the end of a project. An obligation of means was considered as an insufficient guarantee for actual delivery. Some university administrations of the participating research groups initially were not so keen of a contract based on an obligation of results. But the universities benefitted from a reduced administrative overhead imposed by STEVIN: the result counted, not how the means were spent (human resources, equipment, ...). This implied that if a satisfactory result was

[17] This volume obviously addresses this recommendation.

delivered using less funding, the researchers could keep the difference.[18] The challenge for the programme governance was then to determine and monitor the quality of the results – cf. Sect. 2.3.3.

2.4.2 Managing IPR

As mentioned earlier, the ownership of all the STEVIN results (or foreground knowledge), except for the few open source exceptions, eventually went to the NTU. Where needed, extra agreements were concluded regarding background knowledge. The main idea was that all STEVIN results had to be made available and re-usable for academia and industry in the Low Countries. Centralising the ownership within one organisation (which has as its mission to promote and support the Dutch language on behalf of the two funding governments) was seen by the HLT board a good guarantee that STEVIN results would become easily accessible and available for wider distribution. Therefore, an dedicated agency for resource maintenance, distribution and promotion, c.q. the HLT Agency, was created earlier on (cf. Chap. 21, p. 381).

Since it was quite unusual for researchers to have to transfer the ownership of their foreground results, some voiced their discontentment and preference for open source variants, in particular when software[19] was involved. Even if at the start of STEVIN, IPR problems did arise as the HLT board and the programme committee had seriously underestimated the complexity of IPR issues, at the end of STEVIN it became clear that, on the international level, infrastructure projects, resource organisations and even policy makers look enviously at STEVIN as all the IPR rights were legally conclusively cleared (including material of privately hold editing houses), template licence agreements were drafted and ownership was centralised. It still happens all too often that researchers are allowed to use material from commercial third parties only for the specific purpose and duration of a single project. In the case of STEVIN, this limitation, to a very large extent, does not apply thanks to (sometimes time consuming) negotiations and solid agreements with providers. The HLT Agency is now responsible for maintaining and distributing the STEVIN materials and for concluding licence agreements on behalf of the NTU.

From [16], it appears that e.g., open source licences may end up less attractive and more complex for resource distributing agencies or initiatives than initially foreseen. In any case, as the NTU is the proprietor of the STEVIN results, except for some open source materials, any possible way of distribution and exploitation can be applied. For example, a synonym list resulting from the Cornetto (cf. Chap. 10,

[18] One project used this "left-over" money to co-fund the writing of the history of HLT for Dutch in the Low Countries [24].

[19] HLT software is indeed much more difficult to maintain by "less specialised" people (of the HLT Agency). Corpora are easier to maintain in that respect.

p. 165) project could be released as open source to be included in the Dutch "language pack" for Firefox, OpenOffice and Chrome distributed by the OpenTaal organisation.[20] For companies interested in using the Autonomata grapheme-to-phoneme converter, a royalty scheme was set up.

2.5 Conclusion

According to Technopolis group, the general targets of the STEVIN programme have been reached to a (very) large extent. As a sufficient number of high quality project proposals was funded by STEVIN, a substantial strengthening of the digital language infrastructure for Dutch was achieved. The quality of research within STEVIN was, in general, good, albeit not cutting edge. This can be attributed to the nature of the projects (in particular when addressing the BLARK for Dutch) being less apt for high impact publications. Another strong point of STEVIN was the funding of application oriented projects as these demonstrate the potentialities of HLTD to industry and the general public. It resulted in a network with strong ties between academia and industry that is beneficial for future utilisation of the STEVIN results. Some adaptations in the programme governance structure, more interaction with other similar (inter)national R&D programmes, and a better clarification of the role of open source were recommended by the evaluators for a future programme. All in all, they qualify STEVIN as a successful cross-border R&D programme.

Technopolis group recommends to organise a follow-up programme again as a combination of different types of R&D within the same programme: even if focusing more on application-oriented research and demonstrator projects (and thus strengthening the participation of industry and software integrators in the programme), other types of research (e.g., basic research) should not be overlooked. Health care, education, e-government, safety and cultural heritage are cited as potentially interesting application domains for follow-up R&D activities.

According to the evaluator, the Dutch-Flemish HLT community has been able to retain their top position in the international HLT community thanks to STEVIN, which prepared them for a leading position in the European CLARIN endeavour.

Acknowledgements We thank our colleagues of the Nederlandse Taalunie, the HLT board, the HLT Agency, and members of the various related working groups as well as the programme office for their committed collaboration. The reviewers are to be acknowledged for their comments. STEVIN results are presented on the STEVIN web site (www.stevin-tst.org/etalage) and are available via the HLT Agency (www.tst-centrale.org).

[20] http://www.opentaal.org/english

References

1. Akkermans, J., Hochheimer-Richheimer, D., van Rij, I.: Nulmeting STEVIN: startpunt voor bepaling effect STEVIN-programma. Technical report, M&I Partners, Amersfoort (2007) (in Dutch)
2. Beeken, J., Dewallef, E., D'Halleweyn, E.: A platform for Dutch in human language technologies. In: Proceedings of LREC 2000, pp. 63–66, Athens (2000)
3. Binnenpoorte, D., Cucchiarini, C., D'Halleweyn, E., Sturm, J., de Vriend, F.: Towards a roadmap for human language technologies: Dutch-Flemish experience. In: Proceedings of LREC2002, Las Palmas (2002)
4. Binnenpoorte, D., de Vriend, F., Sturm, J., Daelemans, W., Strik, H., Cucchiarini, C.: A field survey for establishing priorities in the development of HLT resources for Dutch. In: Proceedings of the Third International Language Resources and Evaluation Conference (LREC2002), Las Palmas (2002)
5. Boersma, P.: PRAAT, a system for doing phonetics by computer. Glot Int. **5:9/10**, 341–345 (2001)
6. Bouma, G., Schuurman, I.: De positie van het Nederlands in taal- en spraaktechnologie. Technical report, Nederlandse Taalunie (1998)
7. Bouma, G., Schuurman, I.: Intergovernmental language policy for Dutch and the language and speech technology infrastructure. In: Rubio, A., Gallardo, N., Castro, R., Tejada A. (eds.) Proceedings of the First International Conference on Language Resources and Evaluation, Granada, pp. 509–513 (1998)
8. Daelemans, W., Binnenpoorte, D., de Vriend, F., Sturm, J., Strik, H., Cucchiarini, C.: Establishing priorities in the development of HLT resources: the Dutch-Flemish experience. In: Daelemans, W., du Plessis, T., Snyman, C., Teck, L. (eds.) Multilingualism and electronic language management: proceedings of the 4th International MIDP Colloquium, pp. 9–23. Van Schaik, Pretoria (2005)
9. Deuten, J., Mostert, B., Nooijen, A., van der Veen, G., Zuijdam, F.: Eindevaluatie STEVIN-programma. Technical report, The Technopolis Group (2010)
10. Dewallef, E.: Language Engineering in Flanders. Ministry of the Flemish Community, Brussels (1998)
11. D'Halleweyn, E., Odijk, J., Teunissen, L., Cucchiarini, C.: The Dutch-Flemish HLT Programme STEVIN: essential speech and language technology resources. In: Proceedings of LREC 2006, pp. 761–766, Genoa (2006)
12. Goetschalckx, J., Cucchiarini, C., Van Hoorde, J.: Machine translation for Dutch: the NL-Translex project why machine translation? In: Temmerman, R., Lutjeharms, M. (eds.) Proceedings of the International Colloquium Trends in Special Language and Language Technology, Brussels, pp. 261–280. Standaard Editions, Antwerp (2001)
13. Joscelyne, A., Lockwood, R., Euromap Language Technologies: Benchmarking HLT Progress in Europe. Center for Sprogteknologi, Copenhagen (2003)
14. Kessens, J., van Leeuwen, D.: N-Best: the northern and southern Dutch evaluation of speech recognition technology. In: Proceedings of Interspeech 2007, Antwerp, pp. 1354–1357 (2007)
15. Krauwer, S.: The Basic Language Resource Kit (BLARK) as the first milestone for the Language Resources Roadmap. In: Proceedings of the International Workshop Speech and Computer (2003). http://www.elsnet.org/dox/krauwer-specom2003.pdf
16. Oksanen, V., Lindén, K., Westerlund, H.: Laundry symbols and license management: practical considerations for the distribution of LRs based on experiences from CLARIN. In: Proceedings

of the Seventh International Language Resources and Evaluation (LREC'10), Marrakech (2010)
17. Oostdijk, N.: The design of the Spoken Dutch Corpus. In: New Frontiers of Corpus Research, pp. 105–112. Rodopi, Amsterdam/New York (2002)
18. Raw, A., Vandecapelle, B., Van Eynde, F.: Eurotra: an overview. Interface. J. Appl. Linguist. **3**(1), 5–32 (1988)
19. Spyns, P., D'Halleweyn, E.: Flemish-Dutch HLT policy: evolving to new forms of collaboration. In: Calzolari, N., Mouradi, A., et al. (eds.) Proceedings of the 7th Conference on Language Resources and Evaluation (LREC10), pp. 2855–2862. ELRA, Valletta (2010)
20. Spyns, P., D'Halleweyn, E.: Joint research coordination and programming for HLT for Dutch in the Low Countries. J. Linguist. Resour. Eval. (2013, under revision)
21. Spyns, P., D'Halleweyn, E.: Smooth sailing for STEVIN. In: Calzolari, N., Mouradi, A., et al. (eds.) Proceedings of the 8th Conference on Language Resources and Evaluation (LREC12), pp. 1021–1028. ELRA, La Valletta (2012)
22. Spyns, P., D'Halleweyn, E., Cucchiarini, C.: The Dutch-Flemish comprehensive approach to HLT stimulation and innovation: STEVIN, HLT Agency and beyond. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), Marrakech, pp. 1511–1517 (2008)
23. Strik, H., Daelemans, W., Binnenpoorte, D., Sturm, J., de Vriend, F., Cucchiarini, C.: Dutch HLT resources: from BLARK to priority lists. In: Proceedings of ICSLP, Denver (2002)
24. van der Beek, L.: Van rekenmachine tot taalautomaat in de Lage Landen, Groningen (2011). (in Dutch), http://www.let.rug.nl/vannoord/TST-Geschiedenis/
25. van Veenendaal, R., van Eerten, L., Cucchiarini, C.: The Flemish-Dutch HLT Agency: a comprehensive approach to language resources lifecycle management & sustainability for the Dutch language. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), La Valetta (2010)
26. Váradi, T., Krauwer, S., Wittenburg, P., Wynne, M., Koskenniemi, K.: CLARIN: common language resources and technology infrastructure. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), Marrakech, pp. 1244–1248 (2008)

Part II
HLT Resource-Project Related Papers

Chapter 3
The JASMIN Speech Corpus: Recordings of Children, Non-natives and Elderly People

Catia Cucchiarini and Hugo Van hamme

3.1 Introduction

Large speech corpora (LSC) constitute an indispensable resource for conducting research in speech processing and for developing real-life speech applications. The need for such resources is now generally recognised and large, annotated speech corpora are becoming available for various languages. Other than the term "large" probably suggests, all these corpora are inevitably limited. The limitations are imposed by the fact that LSC require much effort and are therefore very expensive. For these reasons, important choices have to be made when compiling an LSC in order to achieve a corpus design that guarantees maximum functionality for the budget available.

In March 2004 the Spoken Dutch Corpus (Corpus Gesproken Nederlands; CGN) became available, a corpus of about nine million words that constitutes a plausible sample of standard Dutch as spoken in the Netherlands and Flanders and contains various annotation layers. The design of this corpus was guided by a number of considerations. In order to meet as many requirements as possible, it was decided to limit the CGN to the speech of adult, native speakers of Dutch in the Netherlands and Flanders.

The rapid developments in Information Society and the ensuing proliferation of computer services in support of our daily activities stress the importance of CGN for developing such services for Dutch at reasonable costs, thus removing the language barrier for many citizens. Familiar examples of Human Language Technology

C. Cucchiarini (✉)
CLST, Radboud University, Nijmegen, The Netherlands
e-mail: c.cucchiarini@let.ru.nl

H. Van hamme
ESAT Department, Katholieke Universiteit Leuven, Leuven, Belgium
e-mail: hugo.vanhamme@esat.kuleuven.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_3, © The Author(s) 2013

(HLT) applications are dictation systems and call-centre-based applications such as telephone transaction systems and information systems that use automatic speech recognition instead of a keyboard or a keypad, such as in-car navigation systems and miniaturised personal assistants. Furthermore, multilingual access interfaces and cross-lingual speech applications in which people can communicate with each other even though they speak different languages are now being developed, i.e. for telephone reservation systems and voice portals. As embedded technology, HLT will have a crucial role in next-generation products and services that replace information processing methods typical of the desktop computing generation. The advent of ambient intelligence will make it possible for humans to interact with ubiquitous computing devices in a seamless and more natural way. Finally, in a world increasingly dominated by knowledge and information, learning will become a lifelong endeavour and HLT applications will become indispensable in favouring remote access and interaction with (virtual) tutors.

3.2 Potential Users of HLT Applications

The fact that CGN is restricted to the speech of adult, native speakers of Dutch in the Netherlands and Flanders, limits its usability for developing HLT applications that must be used by children, non-natives and elderly people. This is undesirable, as these groups also need to communicate with other citizens, administration, enterprises and services and should in principle be able to benefit from HLT-based computer services that are available for the rest of the population. In addition, all three social groups are potential users of HLT applications specially tailored for children, non-natives and elderly people, which would considerably increase their opportunities and their participation in our society.

In the case of children, HLT applications have an important role to play in education and in entertainment [13]. For certain applications, such as internet access and interactive learning, speech technology provides an alternative modality that may be better suited for children compared to the usual keyboard and mouse access. In other applications, such as Computer Assisted Language Learning (CALL) or computer-based interactive reading tutors [9], speech and language technology is the key enabling technology.

The increasing mobility and consequent migration of workers to the Netherlands and Flanders have resulted in growing numbers of non-native speakers of Dutch that have to function in a Dutch-speaking society. For them, HLT applications can be relevant in two respects: to guarantee their participation in the Information Society and to promote their integration in society by facilitating their acquisition of the Dutch language. When talking about the information society, authorities and policy makers put special emphasis on aspects such as empowerment, inclusion, and elimination of cultural and social barriers. This implies that the information society should be open to all citizens, also those who are not mother tongue speakers of Dutch. To guarantee that also non-native speakers of Dutch can participate in

the information society it is necessary that all sorts of services and applications, for instance those mentioned in the previous section, be available for them too. The teaching of Dutch as a second language (L2) is high on the political agenda, both in the Netherlands and in Flanders, because it is considered to be the key to successful integration. In the last 30 years the Dutch and the Flemish governments have spent billions of euros on Dutch L2 teaching to non-natives. Despite these huge efforts, the results are not always satisfactory and experiments are now being conducted with new methods and new media, to try and improve the quality of Dutch L2 teaching. For example, CALL systems that make use of advanced HLT techniques seem to offer new perspectives. These systems can offer extra learning time and material, specific feedback on individual errors and the possibility to simulate realistic interaction in a private and stress-free environment.

Owing to the increase in average life expectancy, our society has to cope with a growing aged population and government and commercial organisations are concerned about how to meet the needs of this increasing group of older adults and to guarantee independent aging as much as possible. Technology, and in particular, HLT applications, can help in providing assistance to older individuals who want to maintain independence and quality of life. Among the consequences of aging are declines in motor, sensory and cognitive capabilities. HLT can be employed in developing assistive devices that compensate for these diminished capabilities. For instance, it is possible to compensate for motor or sensory deficiencies by developing devices for control of the home environment through spoken commands. Cognitive aging often results in a decline in working memory, online reasoning, and the ability to attend to more than one source of information. Technology can compensate for cognitive dysfunctions either by facilitating information processing or by supporting functions such as planning, task sequencing, managing prescription drug regimens, prioritisation and problem solving. The applications can vary from reminder systems to interactive robotic assistants [8, 10, 12, 18].

3.3 The Need for Dedicated Corpora

Although it is obvious that speech-based services are of social and economic interest to youngsters, seniors and foreigners at the moment such applications are difficult to realise. As a matter of fact, speech recognisers that are optimised for adult speech are not suitable for handling speech of children, non-natives and elderly people [3, 6, 13, 15, 20]. The much lower performance achieved with children speech has to do with differences in vocal tract size and fundamental frequency, with pronunciation problems and different vocabulary, and with increased variability within speakers as well as among speakers. In the ASR community, it has long been known that the differences between native and non-native speech are so extensive as to degrade ASR performance considerably [20]. As a consequence, considerable efforts have been spent in trying to understand the reasons for this poor performance and in finding appropriate solutions. Research into automatic speech recognition of

elderly speech has shown that performance degrades considerably for people above the age of 70 [3]. This deterioration in performance can be ascribed to different spectral and pronunciation patterns that result from a degradation of the internal control loops of the articulatory system and from changes in the size and periodicity of the glottal pulses.

Although the performance disadvantage for children, seniors and non-natives can be explained to some extent, there is much that is not well understood. But in the past it has been difficult to conduct research aimed at explaining the difference because of the lack of suitable corpora.

Problems in ASR for children, elderly and non-natives are generally approached with standard adaptation procedures [3, 13, 15, 20]. Although these do improve performance, straightforward adaptation does not bring the performance to the same level as what can be obtained with adult native speech. Perhaps more importantly, straightforward adaptation does not yield much insight into the fundamental causes of the ASR problems. An analysis of turn taking and interaction patterns in the face-to-face and telephone dialogues that was carried out within the COMIC project (http://www.hcrc.ed.ac.uk/comic/documents) showed that these are fundamentally different from the best we can do at this moment in human-computer interaction.

Humans handle misunderstandings and recognition errors seemingly without effort, and that capability appears to be essential for maintaining a fluent conversation. Automatic systems have only very limited capabilities for detecting that their human interlocutor does not fully understand prompts and responses. Experience with developing voice operated information systems has revealed a lack of knowledge about the specific behaviour that people exhibit when they have to interact with automatic systems, especially when the latter do not understand what the user says. For instance, it turns out that people do not answer the questions posed by the machine immediately, but first think about what to say and to take time they either start repeating the question, or produce all sorts of hesitations and disfluencies. In addition, if the computer does not understand them, they start speaking more loudly, or modify their pronunciation in an attempt to be more understandable with the result that their speech deviates even more from what the computer expects. The problems experienced in developing spoken dialogs with machines are compounded when the users come from sections of the population not represented in the corpora used for training the ASR systems, typically children, non-natives and elderly people [13, 15]. Also in spoken human-machine interaction, scientific and technological progress is hampered by the lack of appropriate corpora.

3.4 JASMIN-CGN: Aim of the Project

It is for the reasons mentioned above that within the framework of the Dutch-Flemish programme STEVIN [1] the project JASMIN-CGN was started, which was aimed at the compilation of a corpus of contemporary Dutch as spoken by children of different age groups, elderly people, and non-natives with different

mother tongues in the Netherlands and Flanders. The JASMIN-CGN project was carried out by a Dutch-Flemish consortium made up of two academic institutions (RU Nijmegen, CLST, C. Cucchiarini and KU Leuven, ESAT, H. Van hamme) and TalkingHome, (F. Smits) a company that, at the time, developed speech controlled applications for health care. The JASMIN-CGN project aimed at realising an extension of the Spoken Dutch Corpus (CGN) along three dimensions. First, by collecting a corpus of contemporary Dutch as spoken by children of different age groups, elderly people and non-natives with different mother tongues, an extension along the age and mother tongue dimensions was achieved. In addition, we collected speech material in a communication setting that was not envisaged in the CGN: human-machine interaction.

3.5 Material and Methods

The three dimensions mentioned above are reflected in the corpus as five user groups: native primary school pupils, native secondary school students, non-native children, non-native adults and senior citizens. For all groups of speakers 'gender' was adopted as a selection variable. In addition, 'region of origin' and 'age' constituted variables in selecting native speakers. Finally, the selection of non-natives was also based on variables such as 'mother tongue', 'proficiency level in Dutch' and 'age'.

3.5.1 Speaker Selection

For the selection of speakers we have taken the following variables into account: region of origin (Flanders or the Netherlands), nativeness (native as opposed to non-native speakers), dialect region (in the case of native speakers), age, gender and proficiency level in Dutch (in the case of non-native speakers).

3.5.1.1 Region of Origin

We distinguished two regions: Flanders (FL) and the Netherlands (NL) and we tried to collect one third of the speech material from speakers in Flanders and two thirds from speakers in the Netherlands.

3.5.1.2 Nativeness

In each of the two regions, three groups of speakers consisted of native speakers of Dutch and two of non-native speakers. For native and non-native speakers different selection criteria were applied, as will be explained below.

3.5.1.3 Dialect Region

Native speakers, on the other hand, were divided in groups on the basis of the dialect region they belong to. A person is said to belong to a certain dialect region if (s)he has lived in that region between the ages of 3 and 18 and if (s)he has not moved out of that region more than 3 years before the time of the recording.

Within the native speaker categories we strived for a balanced distribution of speakers across the four regions (one core, one transitional and two peripheral regions) that we distinguished in the Netherlands and Flanders in the sense that we organised recruiting campaigns in each of the regions. However, we did not balance strictly for this criterion, i.e. speakers were not rejected because of it.

For non-native speakers, dialect region did not constitute a selection variable, since the regional dialect or variety of Dutch is not expected to have a significant influence on their pronunciation. However, we did notice a posteriori that the more proficient non-native children do exhibit dialectal influence (especially in Flanders due to the recruitment).

3.5.1.4 Mother Tongue

Since the JASMIN-CGN corpus was collected for the aim of facilitating the development of speech-based applications for children, non-natives and elderly people, special attention was paid to selecting and recruiting speakers belonging to the group of potential users of such applications. In the case of non-native speakers the applications we had in mind were especially language learning applications because there is considerable demand for CALL (Computer Assisted Language Learning) products that can help making Dutch as a second language (L2) education more efficient. In selecting non-native speakers, mother tongue constituted an important variable because certain mother tongue groups are more represented than others in the Netherlands and Flanders. For instance, for Flanders we opted for Francophone speakers since they form a significant fraction of the population in Flemish schools, especially (but not exclusively) in major cities. A language learning application could address the school's concerns about the impacts on the level of the Dutch class. For adults, CALL applications can be useful for social promotion and integration and for complying with the bilingualism requirements associated with many jobs. Often, the Francophone population has foreign roots and we hence decided to also allow speakers living in a Francophone environment but whose first language is not French.

In the Netherlands, on the other hand, this type of choice turned out to be less straightforward and even subject to change over time. The original idea was to select speakers with Turkish and Moroccan Arabic as their mother tongue, to be recruited in regional education centres where they follow courses in Dutch L2. This choice was based on the fact that Turks and Moroccans constituted two of the four most substantial minority groups [5], the other two being people from Surinam and the Dutch Antilles who generally speak Dutch and do not

have to learn it when they immigrate to the Netherlands. However, it turned out that it was very difficult and time-consuming to recruit exclusively Turkish and Moroccan speakers because Dutch L2 classes at the time of recruiting contained more varied groups of learners. This was partly induced by a new immigration law that envisaged new obligations with respect to learning Dutch for people from outside the EU. This led to considerable changes which clearly had an impact on the whole Dutch L2 education landscape. As a consequennce, it was no longer so straightforward to imagine that only one or two mother tongue groups would be the most obvious candidates for using CALL and speech-based applications. After various consultations with experts in the field, we decided not to limit the selection of non-natives to Turkish and Moroccan speakers and opted for a miscellaneous group that more realistically reflects the situation in Dutch L2 classes.

3.5.1.5 Proficiency in Dutch

Since an important aim in collecting non-native speech material is that of developing language learning applications for education in Dutch L2, we consulted various experts in the field to find out for which proficiency level such applications are most needed. It turned out that for the lowest levels of the Common European Framework (CEF), namely A1, A2 or B1 there is relatively little material and that ASR-based applications would be very welcome. For this reason, we chose to record speech from adult Dutch L2 learners at these lower proficiency levels.

For children, the current class (grade) they are in was maintained as a selection criterion. So although in this case proficiency was not really a selection criterion, it is correlated with grade to a certain extent.

3.5.1.6 Speaker Age

Age was used as a variable in selecting both native and non-native speakers. For the native speakers we distinguished three age groups not represented in the CGN corpus:

- Children between 7 and 11
- Children between 12 and 16
- Native adults of 65 and above

For the non-native speakers two groups were distinguished:

- Children between 7 and 16
- Adults between 18 and 60.

3.5.1.7 Speaker Gender

In the five age groups of speakers we strived to obtain a balanced distribution between male and female speakers.

3.5.2 Speech Modalities

In order to obtain a relatively representative and balanced corpus we decided to record about 12 min of speech from each speaker. About 50 % of the material would consist of read speech material and 50 % of extemporaneous speech produced in human-machine dialogues.

3.5.2.1 Read Speech

About half of the material to be recorded from each speaker in this corpus consists of read speech. For this purpose we used sets of phonetically rich sentences and stories or general texts to be read aloud. Particular demands on the texts to be selected were imposed by the fact that we had to record read speech of children and non-natives.

Children in the age group 7–12 cannot be expected to be able to read a text of arbitrary level of difficulty. In many elementary schools in the Netherlands and Flanders children learning to read are first exposed to a considerable amount of explicit phonics instruction which is aimed at teaching them the basic structure of written language by showing the relationship between graphemes and phonemes [26]. A much used method for this purpose is the reading program Veilig Leren Lezen [11]. In this program children learn to read texts of increasing difficulty levels, with respect to text structure, vocabulary and length of words and sentences. The texts are ordered according to reading level and they vary from Level 1 up to Level 9. In line with this practice in schools, we selected texts of the nine different reading levels from books that belong to the reading programme Veilig Leren Lezen.

For the non-native speakers we selected appropriate texts from a widely used method for learning Dutch as a second language, Codes 1 and 2, from Thieme Meulenhoff Publishers. The texts were selected as to be suitable for learners with CEF levels A1 and A2.

3.5.2.2 Human-Machine Dialogues

A Wizard-of-Oz-based platform was developed for recording speech in the human-machine interaction mode. The human-machine dialogues are designed such that the wizard can intervene when the dialogue goes out of hand. In addition, the wizard can simulate recognition errors by saying, for instance: “Sorry, I did not

understand you", or "Sorry, I could not hear you" so as to elicit some of the typical phenomena of human-machine interaction that are known to be problematic in the development of spoken dialogue systems. Before designing the dialogues we drew up a list of phenomena that should be elicited such as hyperarticulation, syllable lengthening, shouting, stress shift, restarts, filled pauses, silent pauses, self talk, talking to the machine, repetitions, prompt/question repeating and paraphrasing. We then considered which speaker moods could cause the various phenomena and identified three relevant states of mind: (1) confusion, (2) hesitation and (3) frustration. If the speaker is confused or puzzled, (s)he is likely to start complaining about the fact that (s)he does not understand what to do. Consequently, (s)he will probably start talking to him/herself or to the machine. Filled pauses, silent pauses, repetitions, lengthening and restarts are likely to be produced when the speaker has doubts about what to do next and looks for ways of taking time. So hesitation is probably the state of mind that causes these phenomena. Finally, phenomena such as hyperarticulation, syllable lengthening, syllable insertion, shouting, stress shift and self talk probably result when speakers get frustrated. As is clear from this characterisation, certain phenomena can be caused by more than one state of mind, like self talk that can result either from confusion or from frustration.

The challenge in designing the dialogues was then how to induce these states of mind in the speakers, to cause them to produce the phenomena required. We have achieved this by asking unclear questions, increasing the cognitive load of the speaker by asking more difficult questions, or by simulating machine recognition errors. Different dialogues were developed for the different speaker groups. To be more precise, the structure was similar for all the dialogues, but the topics and the questions were different.

3.5.3 Collecting Speech Material

3.5.3.1 Speaker Recruitment

Different recruitment strategies were applied for the five speaker groups. The most efficient way to recruit children was to approach them through schools. However, this was difficult because schools are reluctant to participate in individual projects owing to a general lack of time. In fact this was anticipated and the original plan was to recruit children through pedagogical research institutes that have regular access to schools for various experiments. Unfortunately, this form of mediation turned out not to work because pedagogical institutes give priority to their own projects. So, eventually, schools were contacted directly and recruiting children turned out to be much more time-consuming than we had envisaged.

In Flanders, most recordings in schools were organised in collaboration with the school management teams. A small fraction of the data were recorded at summer recreational activities for primary school children ("speelpleinwerking").

The elderly people were recruited through retirement homes and elderly care homes. In Flanders older adults were also recruited through a Third Age University. In the Netherlands non-native children were recruited through special schools which offer specific Dutch courses for immigrant children (Internationale Schakelklassen). In Flanders the non-native children were primarily recruited in regular schools. In major cities and close to the language border a significant proportion of pupils speak only French at home, but attend Flemish schools. The level of proficiency is very dependent on the individual and the age. A second source of speakers was a school with special programs for recent immigrants. Non-native adults were recruited through language schools that offer Dutch courses for foreigners. Several schools (in the Netherlands: Regionale Opleidingscentra, ROCs – in Flanders: Centra voor Volwassenen Onderwijs, CVOs) were invited to participate. Through these schools we managed to contact non-native speakers with the appropriate levels of linguistic skills. Specific organisations for foreigners were also contacted to find enough speakers when recruitment through the schools failed.

All speakers received a small compensation for participating in the recordings in the form of a cinema ticket or a coupon for a bookstore or a toy store.

3.5.3.2 Recordings

To record read speech, the speakers were asked to read texts that appeared on the screen. To elicit speech in the human-machine interaction modality, on the other hand, the speakers were asked to have a dialogue with the computer. They were asked questions that they could also read on the screen and they had received instructions that they could answer these questions freely and that they could speak as long as they wanted.

The recordings were made on location in schools and retirement homes. We always tried to obtain a quiet room for the recordings. Nevertheless, background noise and reverberation could not always be prevented.

The recording platform consisted of four components: the microphone, the amplifier, the soundcard and the recording software. We used a Sennheiser 835 cardoid microphone to limit the impact of ambient sound. The amplifier was integrated in the soundcard (M-audio) and contained all options for adjusting gain and phantom power. Resolution was 16 bit, which was considered sufficient according to the CGN specifications. The microphone and the amplifier were separated from the PC, so as to avoid interference between the power supply and the recordings.

Elicitation techniques and recording platform were specifically developed for the JASMIN-CGN project because one of the aims was to record speech in the human-machine-interaction modality. The recordings are stereo, as both the machine output and the speaker output were recorded.

Table 3.1 Amount of speech material and number of speakers per speaker group. The numbers between round brackets are the number of female participants in each group

Speaker group	NL	FL	NL(F)	FL(F)
Native primary school pupils between 7 and 11	15 h 10 min	7 h 50 min	72 (35)	43 (23)
Native secondary school students between 12 and 16	10 h 59 min	8 h 01 min	63 (31)	44 (22)
Non-native children between 7 and 16	12 h 34 min	9 h 15 min	53 (28)	52 (25)
Non-native adults	15 h 01 min	8 h 02 min	46 (28)	30 (19)
Native adults above 65	16 h 22 min	8 h 26 min	68 (45)	38 (22)
Total	70 h 06 min	41 h 34 min	302 (167)	207 (111)

3.6 Results

3.6.1 Speech Files

In total 111 h and 40 min of speech were collected divided over the different speaker groups as shown in Table 3.1. The corpus documentation contains further details about the speakers (exact age, native language, proficiency in Dutch, gender, place of birth, ...). The samples were stored in 16 bit linear PCM form in a Microsoft Wave Format. The sample frequency is 16 kHz for all recordings. Each recording contains two channels: the output from the TTS system (dialogues) and the microphone recording. Notice that the microphone signal also contains the TTS signal through the acoustic path from the loudspeakers to the microphone.

About 50 % of the material is read speech and 50 % extemporaneous speech recorded in the human-machine interaction modality (HMI).

3.6.2 Orthographic Annotations

All speech recordings were orthographically transcribed manually according to the same conventions adopted in CGN and using the same tool: PRAAT [2]. Since this corpus also contains speech by non-native speakers, special conventions were required, for instance, for transcribing words realised with non-native pronunciation. Orthographic transcriptions were made by one transcriber and checked by a second transcriber who listened to the sound files, checked whether the orthographic transcription was correct and, if necessary, improved the transcription. A spelling check was also carried out according to the latest version of the Dutch spelling [14]. A final check on the quality of the orthographic transcription was carried out by running the program 'orttool'. This program, which was developed for CGN but

was not further disseminated, checks whether markers and blanks have been placed correctly and, if necessary, improves the transcription.

The speech material recorded in the Netherlands was also transcribed in the Netherlands, whereas the speech material recorded in the Flanders was transcribed in Flanders. To avoid inconsistencies in the transcription, cross checks were carried out.

3.6.3 Annotations of Human-Machine Interaction Phenomena

A protocol was drawn up for transcribing the HMI phenomena that were elicited in the dialogues. This document can be found in the corpus documentation. The aim of this type of annotation was to indicate these phenomena so that they can be made accessible for all sorts of research and modeling. As in any type of annotation, achieving an acceptable degree of reliability is very important. For this reason in the protocol we identified a list of phenomena that appear to be easily observable and that are amenable to subjective interpretation as little as possible. The following phenomena were transcribed: hyperarticulation, syllable lengthening, shouting, stress shift, restarts, filled pauses, silent pauses, understanding checks, self talk, repetitions, prompt/question repeating and rephrasing. In addition, examples were provided of the manifestation of these phenomena, so as to minimise subjectivity in the annotation.

As for the orthographic transcriptions, the HMI transcriptions were also made by one transcriber and checked by a second transcriber who listened to the sound files, checked whether the transcription was correct and, if necessary, improved it. The speech material recorded in the Netherlands was also transcribed in the Netherlands, whereas the speech material recorded in the Flanders was transcribed in Flanders. To prevent inconsistencies in the transcription, cross checks were carried out.

3.6.4 Phonemic Annotations

It is common knowledge, and the experience gained in CGN confirmed this, that manually generated phonetic transcriptions are very costly. In addition, recent research findings indicate that manually generated phonetic transcriptions are not always of general use and that they can be generated automatically without considerable loss of information [19]. In a project like JASMIN-CGN then an important choice to make is whether the money should be allocated to producing more detailed and more accurate annotations or simply to collecting more speech material. Based on the considerations mentioned above and the limited budget that was available for collecting speech of different groups of speakers, we chose the second option and decided to adopt an automatically generated broad phonetic transcription (using Viterbi alignment).

Given the nature of the data (non-native, different age groups and partly spontaneous), the procedure requires some care. Since the performance of an automatic speech aligner largely depends on the suitability of its acoustic models to model the data set, it was necessary to divide the data into several categories and treat each of those separately. Those categories were chosen such that the data in each could be modelled by a single acoustic model, making a compromise between intra-category variation and training corpus size. Both for Flemish and Dutch data we therefore made the distinction between native children, non-native children, native adults, non-native adults and elderly people.

Deriving an acoustic model for each category was not a straightforward task, since the amount of available data was not always sufficient, especially for the Flemish speakers. In all cases, we started from an *initial* acoustic model and adapted that to each category by mixing in the data on which we needed to align. For children, however, both native and non-native, this solution was not adequate. Since vocal tract parameters change rather drastically during childhood, a further division of the children data according to age at the time of recording was mandatory. We distinguished speakers between 5 and 9 years old, speakers between 10 and 12 years old, and speakers between 13 and 16 years old.

These sets of children data were then used to determine suitable vocal tract length warping factors, in order to apply VTLN (Voice Tract Length Normalisation) [7]. Because of this, data from speakers of all ages could be used in deriving suitable acoustic models for children data. To end up with an acoustic model for each of the ten categories we distinguished in the data, we used four initial acoustic models: Dutch native children (trained on roughly 14 h of JASMIN data), Flemish native children (trained on a separate database), Dutch native adults (trained on CGN) and Flemish native adults (trained on several separate databases). For each category of speakers, a suitable model was derived from one of these initial models by performing a single training pass on it. For instance, to align the Flemish senior speech, a single training pass was performed on the model for Flemish native adult speech using the Flemish senior data.

The quality of the automatic annotation obtained by the speech aligner depends on the quality of the lexicon used. These lexicons should contain as many pronunciation variants for each word as possible for the Viterbi aligner to choose from. For instance, the “n” at the end of a Dutch verb or plural noun is often not pronounced, especially in sloppy speech. The omission of this “n” should be accounted for in the lexicon. The base lexicons were Fonilex for Flemish and CGN for Dutch. Additionally, two pronunciation phenomena, which were not present in CGN, were annotated manually in the JASMIN database: pause in a word, (typically in hesitant speech by non-natives, which was annotated orthographically with “*s” following the word) and foreign pronunciation of a word (marked by a trailing *f). The lexicon for these words was created manually in several iterations of inspection and lexicon adaptation. In general, this leads to an increase in the options the Viterbi aligner can choose from. Further modelling of pronunciation variation is in hard-coded rules as in the CGN. An example of such a rule is vowel substitution due to dialectic or non-native pronunciation.

Quality checks of the automatically generated phonemic transcriptions were carried out by verifying the proposed transcription for three randomly selected files per Region (FL/NL) and category (non-native child, non-native adult, native child and senior) (a total of 24 recordings). Lexicon and cross-word assimilation rules were adapted to minimise the number of errors. Most of the required corrections involved hard/soft pronunciation of the "g" and optional "n" in noun plurals and infinitive forms.

3.6.5 Part-of-Speech Tagging

For all (orthographic) transcriptions, a part of speech (PoS) tagging was made. This was done fully automatically by using the POS tagger that was developed for CGN at ILK/Tilburg University. Accuracy of the automatic tagger was about 97 % on a 10 % sample of CGN [21]. The tagset consists of 316 tags and is extensively described (in Dutch) in [25]. Manual correction of the automatic POS tagging was not envisaged in this project.

3.6.6 External Validation and Distribution

The JASMIN speech corpus was validated by BAS Services at the Phonetics Institute of Munich University against general principles of good practice and the validation specifications provided by the JASMIN consortium. The validation had the following aims:

1. Assess the formal correctness of the data files
2. Assess the correctness of the transcriptions and annotations, specifically the orthographic transcriptions, the automatically generated phonemic transcriptions and the HMI annotations.
3. Indicate to what extent transcriptions and annotations were in line with the guidelines laid down in the corresponding protocols.
4. Determine whether the protocols provided adequate information for users of the corpus.

The validation concerned completeness, formal checks and manual checks of randomly selected samples. Data types covered by this validation were corpus structure, signal files, orthographic, phonetic, POS, HMI events annotation and all English documentation files. Manual checks were carried out by native Dutch and Flemish speakers for the orthographic transcript, the phonetic transcript and the HMI event labelling.

The validation results indicated that the JASMIN corpus was of sufficient quality and received a relatively high score (16/20). In addition, minor errors or inaccuracies signaled during validation were subsequently redressed by the JASMIN consortium

before transferring the JASMIN corpus to the Dutch-Flemish HLT Agency, which is now in charge of its management, maintenance and distribution.

3.7 Discussion

Eventually, the realisation of the JASMIN-CGN corpus has required much more time than was initially envisaged. The lion share of this extra time-investment was taken up by speaker recruiting. We had anticipated that speaker recruiting would be time consuming because, owing to the diversity of the speaker groups, we had to contact primary schools, secondary schools, language schools and retirement homes in different dialect regions in the Netherlands and Flanders. In addition, we knew that schools are often reluctant to participate in external projects. Nevertheless, speaker recruiting turned out to be more problematic than we had expected. Anyway, one lesson we learned is that while talking to user groups one should not only ask them about their wishes, but also about the feasibility of what they suggest.

Another thing that we realised along the way is that very often, especially in schools, various forms of research or screening are carried out for which also speech recordings are made of children or non-native speakers. These speech data could be used not only for the studies for which they were originally collected, but also for further use in HLT. The only problem is that, in general, the researchers in question do not realise that their data could be valuable for other research fields. It would therefore be wise to keep track of such initiatives and try to make good agreements with the researchers in charge to ensure that the recordings are of good quality and that the speakers are asked to give their consent for storing the speech samples in databases to be used for further research, of course with the necessary legal restrictions that the data be made anonymous and be used properly. This would give the opportunity of collecting additional speech material in a very efficient and less expensive way.

3.8 Related Work and Contribution to the State of the Art

Since its completion in 2008, the JASMIN corpus has been employed for research and development in various projects. At CLST in Nijmegen the non-native part of the JASMIN corpus appeared to be particularly useful for different lines of research, as will be explained below. Within the STEVIN programme, the JASMIN corpus has been used in the DISCO project (cf. Chap. 18, p. 323 on the DISCO project).

In DISCO the adult non-native JASMIN speech material was used in combination with the SPRAAK toolkit (cf. Chap. 6, p. 95) in research aimed at optimising automatic speech recognition of low-proficient non-native speakers [23]. In addition, the same JASMIN subcorpus was employed in research on automatic detection of pronunciation errors [22] and in research aimed at developing alternative automatic

measures of pronunciation quality [23]. For these purposes the automatically generated phonemic transcriptions of the adult non-native speech material were manually verified by trained transcribers.

Furthermore, the adult non-native part of the JASMIN corpus also appeared to be particularly suited for studying possible differences in pronunciation error incidence in read and spontaneous non-native speech [24] and for investigating fluency phenomena in read and spontaneous speech samples of one and the same non-native speaker [4]. Finally, the JASMIN adult non-native dialogues were successfully employed to the benefit of research on automatic detection of syntactical errors in non-native utterances [16, 17].

Recently, new research started at the Max Planck Institute in Nijmegen which is aimed at studying aging and the effects of lexical frequencies on speech production. For this purpose the elderly speech of the JASMIN corpus will be employed.

Although the above list indicates that the JASMIN speech corpus has already been used for different investigations, it is clear that its use has so far been relatively limited to researchers that had been involved in its realisation. In a sense this is obvious, because the compilers of the corpus know it in detail and are more able to gauge its potential. However, it seems that more effort should be put in raising awareness among researchers of the availability of these speech data and the possibilities they offer for research and development. This is a challenge for the Dutch-Flemish HLT Agency, which is now in charge of the JASMIN speech corpus and its future lifecycle.

Acknowledgements We are indebted to the publishers Thieme-Meulenhoff and Zwijsen who allowed us to use their texts for the recordings, to A. van den Bosch who allowed us to use the POS tagger, to all the speakers as well as institutions that participated and thus made it possible to collect this corpus and to the people who, at different stages and for different periods, were part of the JASMIN team: Leontine Aul, Andrea Diersen, Joris Driesen, Olga van Herwijnen, Chantal Mülders, August Oostens, Eric Sanders, Maarten Van Segbroeck, Alain Sips, Felix Smits, Koen Snijders, Erik Stegeman and Barry van der Veen.

References

1. http://taalunieversum.org/taal/technologie/stevin/
2. http://www.fon.hum.uva.nl/praat/
3. Anderson, S., Liberman, N., Bernstein, E., Foster, S., Cate, E., Levin, B., Hudson, R.: Recognition of elderly speech and voice-driven document retrieval. In: Proceedings of the ICASSP, Phoenix, USA (1999). Paper 2060
4. Cucchiarini, C., van Doremalen, J., Strik, H.: Fluency in non-native read and spontaneous speech. In: Proceedings of DiSS-LPSS Joint Workshop, Tokyo, Japan, pp. 15–18 (2010)

5. Dagevos, J., Gijsberts, M., van Praag, C.: Rapportage minderheden 2003; Onderwijs, arbeid en sociaal-culturele integratie. SCP-publicatie 2003–13, Sociaal en Cultureel Planbureau, The Hague (2003)
6. D'Arcy, S.M., Wong, L., Russell, M.J.: Recognition of read and spontaneous children's speech using two new corpora. In: Proceedings of ICSLP, Korea pp. 588–591 (2004)
7. Duchateau, J., Wigham, M., Demuynck, K., Van hamme, H.: A flexible recogniser architecture in a reading tutor for children. In: Proceedings of ITRW on Speech Recognition and Intrinsic Variation, Toulouse, France, pp. 59–64 (2006)
8. Ferguson, G., et al.: The medication advisor project: preliminary report. Technical Report 776, CS Department, U. Rochester (2002)
9. Hagen, A., Pellom, B., Cole, R.: Children's speech recognition with application to interactive books and tutors. In: Proceedings of ASRU, St. Thomas, USA, pp. 265–293 (2003)
10. Hans, M., Graf, B., Schraft, R.: Robotic home assistant care-o-bot: past-present-future. In: Proceedings of the IEEE ROMAN, Berlin, pp. 380–385 (2002)
11. Mommers, M., Verhoeven, L., Van der Linden, S.: Veilig Leren Lezen. Zwijsen, Tilburg (1990)
12. Müller, C., Wasinger, R.: Adapting multimodal dialog for the elderly. In: Proceedings of the ABIS-Workshop 2002 on Personalization for the Mobile World, Hannover, Germany, pp. 31–34 (2002)
13. Narayanan, S., Potamianos, A.: Creating conversational interfaces for children. IEEE Trans. Speech Audio Process. **10**(2), 65–78 (2002)
14. Nederlandse Taalunie: Woordenlijst nederlandse taal (2005). http://woordenlijst.org/
15. Raux, A., Langner, B., Black, A., Eskenazi, M.: LET'S GO: improving spoken dialog systems for the elderly and non-natives. In: Proceedings of Eurospeech, Geneva, Switzerland, pp. 753–756 (2003)
16. Strik, H., van de Loo, J., van Doremalen, J., Cucchiarini, C.: Practicing syntax in spoken interaction: automatic detection of syntactic errors in non-native utterances. In: Proceedings of the SLaTE-2010 Workshop, Tokyo, Japan (2010)
17. Strik, H., van Doremalen, J., van de Loo, J., Cucchiarini, C.: Improving asr processing of ungrammatical utterances through grammatical error modeling. In: Proceedings of the SLaTE-2010 Workshop, Venice, Italy (2011)
18. Takahashi, S., Morimoto, T., Maeda, S., Tsuruta, S.: Spoken dialogue system for home health care. In: Proceedings of ICSLP, Denver, USA (2002), pp. 2709–2712
19. Van Bael, C., Binnenpoorte, D., Strik, H., van den Heuvel, H.: Validation of phonetic transcriptions based on recognition performance. In: Proceedings of Eurospeech, Geneva, Switzerland (2003), pp. 1545–1548
20. Van Compernolle, D.: Recognizing speech of goats, wolves, sheep and . . . non-natives. Speech Commun. **35**(1–2), 71–79 (2001)
21. Van den Bosch, A., Schuurman, I., Vandeghinste, V.: Transferring pos-tagging and lemmatzation tools from spoken to written dutch corpus development. In: Proceedings of LREC, Genoa, Italy (2006)
22. van Doremalen, J., Cucchiarini, C., Strik, H.: Automatic detection of vowel pronunciation errors using multiple information sources. In: Proceedings of ASRU, Merano, Italy, (2009), pp. 80–585
23. van Doremalen, J., Cucchiarini, C., Strik, H.: Optimizing automatic speech recognition for low-proficient non-native speakers. EURASIP J. Audio, Speech, Music Process. (2010). http://www.hindawi.com/journals/asmp/2010/973954/
24. van Doremalen, J., Cucchiarini, C., Strik, H.: Phoneme errors in read and spontaneous non-native speech: relevance for capt system development. In: Proceedings of the SLaTE-2010 Workshop, Tokyo, Japan (2010b)
25. Van Eynde, F.: Part of speech tagging en lemmatisering van het corpus gesproken nederlands (2004). http://lands.let.ru.nl/cgn/doc_Dutch/topics/version_.0/annot/pos_tagging/tg_prot.pdf
26. Wentink, H.: From graphemes to syllables. Ph.D. thesis, Nijmegen (1997)

Chapter 4
Resources Developed in the Autonomata Projects

Henk van den Heuvel, Jean-Pierre Martens, Gerrit Bloothooft, Marijn Schraagen, Nanneke Konings, Kristof D'hanens, and Qian Yang

4.1 Introduction

In many modern applications such as directory assistance, name dialing, car navigation, etc. one needs a speech recognizer and/or a speech synthesizer. The former to recognize spoken user commands and the latter to pronounce information found in a database. Both components need phonemic transcriptions of the words to recognize/pronounce, and since many of these words are names, having good automatic phonemic transcription of names is crucial for application development.

A problem, especially in view of the recognition of names, is the existence of different pronunciations of the same name. These pronunciations often depend on the background (mother tongue) of the user. Typical examples are the pronunciation of foreign city names, foreign proper names, etc. The first goal of Autonomata was, therefore, to collect a large number of name pronunciations and to provide manually checked phonetic transcription of these name utterances. Together with meta-data for the speakers, such data is a valuable resource in the research towards a better name recognition.

In order to develop an application, the developer further needs a tool that accepts words/sentences and that returns the phonetic transcriptions of these words/sentences. The second goal of the Autonomata project was to develop a tool that incorporates a state-of-the-art grapheme-to-phoneme convertor (in our case

H. van den Heuvel (✉) · N. Konings
CLST, Radboud University, Nijmegen, The Netherlands
e-mail: H.vandenHeuvel@let.ru.nl

J.-P. Martens · K. D'hanens · Q. Yang
ELIS, Gent University, Gent, Belgium
e-mail: Jean-Pierre.Martens@elis.ugent.be

G. Bloothooft · M. Schraagen
UiL-OTS, Utrecht University, Utrecht, The Netherlands
e-mail: G.Bloothooft@uu.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_4, © The Author(s) 2013

from Nuance), as well as a dedicated phoneme-to-phoneme (p2p) post-processor which can automatically correct some of the mistakes which are being made by the standard g2p. Dedicated p2p post-processors were developed for person names and geographical names.

In the follow-up Autonomata project (Autonomata TOO[1]) the aim was to build a demonstrator version of a Dutch/Flemish Points of Interest (POI) information providing business service, and to investigate new pronunciation modeling technologies that can help to bring the spoken name recognition component of such a service to the required level of accuracy. In order to test the technology for POIs a speech database designed for POI recordings was needed and compiled.

In this contribution we will describe in more detail the four resources briefly sketched above that were developed in Autonomata and in Autonomata Too[2]:

1. The Autonomata Spoken Names Corpus (ASNC)
2. The Autonomata transcription Toolbox
3. The Autonomata P2P converters
4. The Autonomata TOO Spoken POI Corpus

Another contribution in this book (Chap 14, p. 251) will address the research carried out in the Autonomata projects.

4.2 The Autonomata Spoken Names Corpus (ASNC)

4.2.1 Speakers

The ASNC[3] includes spoken utterances of 240 speakers living in the Netherlands (NL) or in Flanders (FL). The speakers were selected along the following dimensions:

1. Main region: 50 % persons living in the Netherlands and 50 % living in Flanders
2. Nativeness: 50 % native speakers of Dutch and 50 % non-native speakers
3. Dialect region of *native* speakers: four dialect regions per main region
4. Mother tongue of *non-native* speakers: three mother tongues per main region
5. Speaker age: one third younger than 18
6. Speaker gender: 50 % male, 50 % female

[1]Too stands for Transfer Of Output. Autonomata TOO used the output of the first project to demonstrate the potential of the technology.

[2]Partners in both projects were Radboud University Nijmegen (CLST), Gent University (ELIS), Utrecht University (UiL-OTS), Nuance, and TeleAtlas. Autonomata lasted from June 2005 to May 2007; Autonomata Too lasted from February 2008 to July 2010.

[3]Section 4.2 is largely based on [4].

We aimed to recruit non-native speakers that still speak their (foreign) mother tongue at home and that have a level A1, A2 or B1 (CEF standard[4]) for Dutch. However, the above strategy appeared to be too restrictive given the limited amount of time there was to finish the speaker recruitment. Another problem was that Flemish schools do not work with the CEF standard. Nevertheless, whenever the CEF information was available, it was recorded and included in the speaker information file.

The 60 non-native speakers in a region were divided into three equally large groups. But since French is obviously an important language in Flanders and far less important in the Netherlands, the division in subgroups has been made differently in the two main regions:

- In **Flanders,** speakers with an **English, French** and **Moroccan** (Arabic) mother tongue were selected.
- In the **Netherlands,** speakers with an **English, Turkish** and **Moroccan** (Arabic) mother tongue were selected.

As foreign speakers mostly live in the big cities and as the dialect region they live in is expected to have only a minor influence on their pronunciation, the dialect region was no selection criterion for these speakers. Native speakers on the other hand were divided in groups on the basis of the dialect region they belong to. A person is said to belong to a certain dialect region if s/he has lived in that region between the ages of 3 and 18 and if s/he has not moved out of that region more than 3 years before the time of the recording. We adopted the same regions that were also used for the creation of the CGN (Spoken Dutch) corpus.[5]

The speaker selection criteria altogether resulted in the speaker categorization shown in Table 4.1.

4.2.2 Recording Material

Each speaker was asked to read 181 proper names and 50 command and control words from a computer screen. The command words are the same for every speaker, but in each region, the names read by a speaker are retrieved from a long list of 1,810 names. These lists were created independently in each region, meaning that there is only a small overlap between the names in the two long lists. Once created, the long list was subdivided in ten mutually exclusive short lists, each containing 181 names: 70 % names that are typical for the region (NL/FL) and 30 % names that are typical for the mother tongues covered by the foreign speakers (10 % for each mother tongue). The typical names for a region were further subdivided in 50 % frequent and 50 % less frequent words.

[4] http://en.wikipedia.org/wiki/Common_European_Framework_of_Reference_for_Languages

[5] http://lands.let.ru.nl/cgn/doc_Dutch/topics/version_1.0/metadata/speakers.htm

Table 4.1 Speaker distribution in the spoken name corpus

Region	Origin	Dialect region
120 Dutch (50 % males)	60 natives	15 WestDutch
		15 Transitional region
		15 Northern
		15 Southern
	60 non-natives	20 English
		20 Turkish
		20 Moroccan
120 Flemish (50 % males)	60 natives	15 Antwerp and Brabant
		15 East Flanders
		15 West Flanders
		15 Limburg
	60 non-natives	20 English
		20 French
		20 Moroccan

For the native speakers we used all ten short lists, meaning that each name is pronounced by six native speakers of a region. For the non-native speakers we worked with only six short lists in order to achieve that the same name occurs three or four times in each non-native subgroup (right column of Table 4.1).

For all languages except Moroccan we selected 25 % person names (each person name consists of a first name and a family name), 35 % street names and 15 % town or city names. We selected more street names than city names because there are – logically – more streets than cities in a country. For the Moroccan names, we chose to select only person names because Dutch speakers will only rarely be confronted with Moroccan geographical names. Furthermore, we adopted the French way of writing for Moroccan names.

Exonyms were not included; meaning that we selected "Lille" instead of "Rijsel". Acronyms for highways (e.g. E40, A12) were not selected either.

We also took care that all different standard elements like street, drive, avenue... are present in a proportional way.

Since first names and family names naturally go together, it was decided to select a first name and a family name of the same language of origin and the same frequency class (in case of typical Dutch names).

Since it may be interesting to investigate whether speaker-specific pronunciation phenomena can be derived to some extent from a restricted set of adaptation data, it was decided to let every speaker also pronounce a list of 50 words that are often encountered in the context of an application and that reveal a sufficient degree of acoustic variability to make the word utterances also suitable for acoustic model adaptation. A list of 50 such words was delivered by Nuance (cf. Table 4.2). It consists of 15 digit sequences and 35 common command and control words.

Table 4.2 Commands and control words included in the ASNC

0 7 9 1	9 0 2 3	sluiten	opnemen	netwerk
3 9 9 4	9 5 6 0	bevestigen	programmeren	infrarood
0 2 8 9	0 1 2 3	controleren	microfoon	instellingen
5 6 9 4	1 6 8 3	help	stop	herhaal
2 3 1 4	7 8 2 6	ga naar	opslaan	opnieuw
7 8 9 0	activeren	aanschakelen	macro	menu
2 2 2 3	annuleren	Nederlands	controlemenu	opties
5 6 7 8	aanpassen	herstarten	status	lijst
9 0 7 4	ga verder	spelling	batterij	Vlaams
3 2 1 5	openen	cijfer	signaalsterkte	Frans

4.2.3 Recording Equipment and Procedure

The speakers were asked to pronounce an item that was displayed in a large font on a computer screen in front of them. Every participant had to read 181 name items (cf. Sect. 4.2.2) and 50 command word items. To simulate the fact that in a real application environment, the user usually has some idea of the name type s/he is going to enter, the participants in our recordings were also given background information about the origin of the names. To that end, the name items were presented per name category: Dutch person names, English person names, Dutch geographical names, etc. The name category was displayed before the first name of that category was prompted.

For the presentation and recording we used software that is commonly used by Nuance for the collection of comparable speech databases.

The microphone was a Shure Beta 54 WBH54 headset supercardoid electret condenser microphone. A compact four Desktop audio mixer from Soundcraft was used as a pre-amplifier. The 80 Hz high-pass filter of the audio mixer was inserted in the input path as a means for reducing low frequency background noise that might be present in the room.

The speech was digitized using an external sound card (VXPocket 440) that was plugged into a laptop. The digital recordings were immediately saved on hard disk. The samples were stored in 16 bit linear PCM form in a Microsoft Wave Format. The sample frequency was 22.05 kHz for all recordings. Before and after every signal there is supposed to be at least 0.5 s of silence (this instruction was not always followed rigorously).

In Flanders, a large part of the recordings were made **in studios** (especially those of non-native speakers and adult speakers), the rest was made **in schools** (those of young speakers and non-natives who take courses in a language center). Recordings in schools may be corrupted by background noise and reverberation. In the Netherlands all recordings were made on location, mostly in schools.

4.2.4 Annotations

Each name token has an orthographical and four broad phonemic transcriptions (cf. Sect. 4.1). Two transcriptions were automatically generated by the Dutch and Flemish versions of the Nuance g2p, respectively. A hand crafted example transcription that is supposed to represent a typical pronunciation of the name in the region of recording was created by a human expert. Finally, an auditory verified transcription was produced by a person with experience in making phonemic transcriptions of speech recordings. All phonemic transcriptions consist of phonemes, word boundaries, syllable boundaries and primary stress markers. The automatically generated transcriptions were converted from the Nuance internal format to the CGN format.[6]

Obviously, the first three transcriptions are the same for all utterances of the same name in one region, and as a consequence, they are provided in the name lists, together with the orthography and the type and language of origin of the name.

The auditory verified transcriptions are specific for each utterance. These transcription files were made in Praat.[7] The annotator could listen to an utterance as many times as s/he wished, and s/he was asked to modify (if necessary) the example transcription that was displayed above the signal. The modification was done according to rules outlined in a phonemic transcription protocol that is distributed together with the corpus.

For the sake of consistency we chose to work with example transcriptions for all names, even though for foreign names spoken by native Dutch/Flemish speakers and Dutch/Flemish names spoken by foreigners these standard transcriptions do not really offer a time gain compared to transcribing from scratch.

4.2.5 Corpus Distribution

The corpus is 9 GB large and is distributed by the Dutch HLT-agency (TST-centrale).[8] The corpus has a rich body of documentation. There is a general documentation file describing all aspects of the corpus construction as well as the format and content of all corpus files. The documentation also contains the phonemic transcription protocol (in Dutch) that was used for the creation of the example transcriptions and the auditory verified transcriptions, as well as a translation of that protocol in English, Also included is a document (in Dutch) describing the internal validation experiments that were carried out in the course of the corpus construction process.

[6] http://lands.let.ru.nl/cgn/doc_English/topics/version1.0/formats/text/fon.htm

[7] http://www.praat.org

[8] http://www.tst-centrale.org/nl/producten/corpora/autonomata-namencorpus/6-33

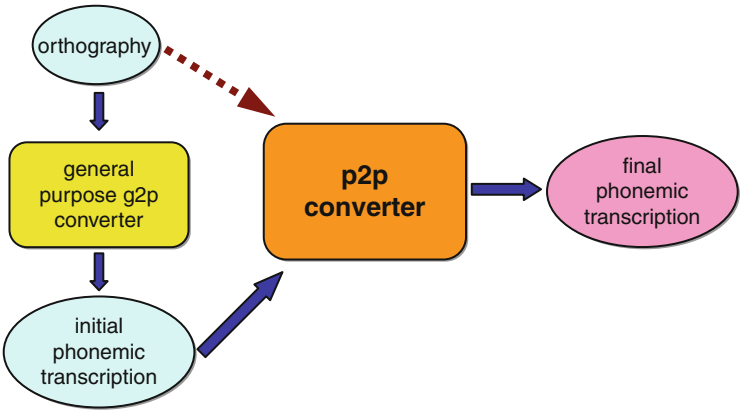


Fig. 4.1 Architecture of a two-step g2p converter

4.3 The Autonomata Transcription Toolbox

This toolset consists of a grapheme-to-phoneme (g2p) transcription tool and a phoneme-to-phoneme (p2p) learning tool.[9] The transcription tool is designed to enrich word lists with detailed phonetic transcriptions. It embeds the state-of-the-art general purpose g2p converters of Nuance (for northern and southern Dutch, English, French and German) and it can upload one or more specialized phoneme-to-phoneme (p2p) converters that were created by the p2p learning tool and that were designed to improve the outputs of the general-purpose g2p converter for names from a specific domain (e.g. street names, POIs, brand names, etc.). The p2p learning tool offers the lexicon developer the means of creating suitable p2p converters from a small lexical database of domain names and their correct transcription (see [4, 6, 7]). The p2p converters can be configured to generate multiple pronunciations with associated probabilities.

4.3.1 A Two-Step g2p Converter Strategy

The general architecture of the proposed two-step g2p conversion system is depicted in Fig. 4.1.

The general-purpose g2p converter creates an initial phonemic transcription which is then corrected by the p2p converter. In order to perform its work, the p2p converter can inspect both the initial phonemic transcription and the orthography of the name it has to process. The heart of the p2p converter is a set of stochastic correction rules, with each rule expressing the following:

[9] This section is largely based on [7].

If a particular phonemic pattern (called the rule input) occurs in the initial phonemic transcription and if the context in which it occurs meets the rule condition, then it may have to be transformed, with a certain firing probability, to an alternative phonemic pattern (called the rule output) in the final transcription.

The rule condition can describe constraints on the identities of the phonemes to the left and the right of the rule input, the stress level of the syllable associated with that input, the position of this syllable in the word, etc. It can also express constraints on the graphemic patterns that gave rise to the rule input and the contextual phonemes.

We distinguish three types of correction rules: (1) stress substitution rules (SS-rules) which replace a stress mark by another (no stress is also considered as a stress mark here), (2) phoneme substitution and deletion rules (PSD-rules) which transform a phonemic pattern into another one (including the empty pattern representing a pattern deletion) and (3) phoneme insertion rules (PI-rules) inserting a phonemic pattern at some position. The linguistic features for describing the context can be different for the respective rule types.

The rewrite rules are implemented in the form of decision trees (DTs). Each DT comprises the rules that apply to a particular rule input. The DTs are learned automatically from training examples by means of machine learning algorithms that were previously applied with success to add pronunciation variants to the lexicon of an automatic speech recognizer.

4.3.2 Learning the Correction Rules

The whole rule learning process is depicted in Fig. 4.2 (cf. also Chap. 14, Sect. 14.2, p. 260).

In general terms, the process is applied to a set of training objects each consisting of an orthography, an initial g2p transcription (called the source transformation), the correct transcription (called the target transcription) and a set of high-level semantic features (e.g. the name type or the language of origin) characterizing the name. Given these training objects, the learning process then proceeds as follows:

1. The objects are supplied to an alignment process incorporating two components: one for lining up the source transcription with the target transcription (sound-to-sound) and one for lining up the source transcription with the orthography (sound-to-letter). These alignments, together with the high-level features are stored in an alignment file.
2. The transformation learner analyzes the alignments and identifies the (focus, output) pairs that are capable of explaining a lot of systematic deviations between the source and the target transcriptions. These pairs define transformations which are stored in a transformation file.
3. The alignment file and the transformation file are supplied to the example generator that locates focus patterns from the transformation file in the source transcriptions, and that generates a file containing the focus, the corresponding

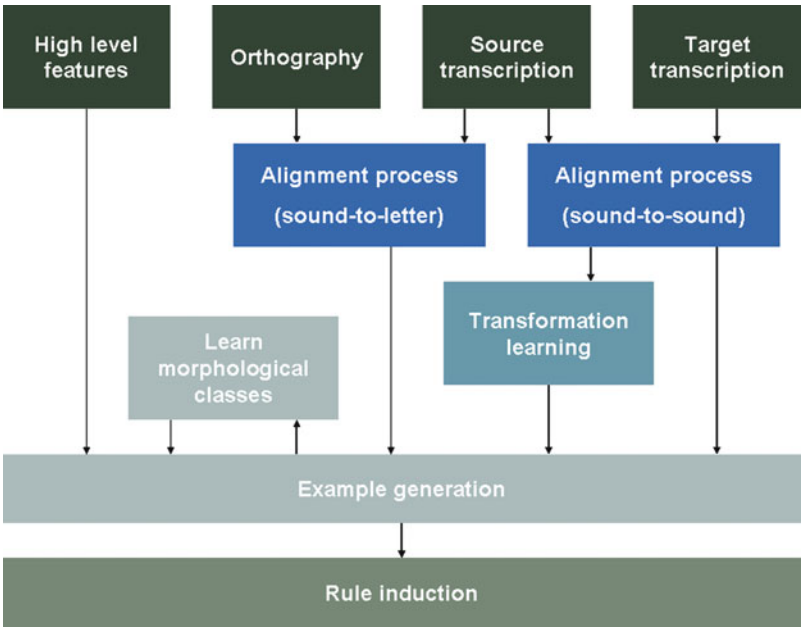


Fig. 4.2 Process for automatically learning of a P2P converter

contextual features and the output for each detected focus pattern. These combinations will serve as the examples from which to train the rules. The example generator also provides statistics about the words whose initial transcription is incorrect, and from these statistics one can create prefix, suffix and syllable sets which define 'morphological' features that can be added to the already mentioned feature set. By running the example generator a second time one creates training examples which also incorporate these 'morphological' features.

4. The example file is finally supplied to the actual rule induction process which automatically constructs a decision tree per focus.

In the subsequent subsections we further elaborate the rule learning process and we also indicate where a manual intervention is possible or desirable. For a more in-depth discussion of the process, the reader is referred to [1] and the documentation provided with the software.

4.3.2.1 Alignment

As indicated before, the alignment process performs a sound-to-letter (or phoneme-to-grapheme) alignment between the source transcription and the orthography, and a sound-to-sound (or phoneme-to-phoneme) alignment between the source and the target phonemic transcription. By replacing every space in the orthography by the

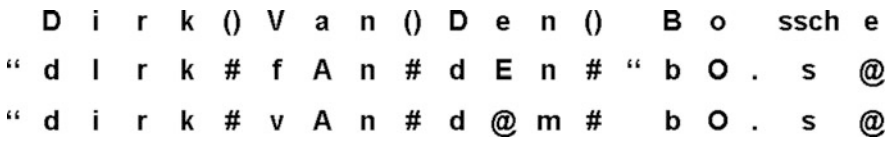


Fig. 4.3 Alignment of the orthography (*top*), the source transcription (*mid*) and the target transcription (*bottom*) of the person name *Dirk Van Den Bossche*

symbol “()”, one can visualize both alignments together in the form of a matrix (cf. Fig. 4.3). The rows subsequently represent the orthography (row 1), the source transcription (row 2) and the target transcription (row 3). The alignment between a source transcription and a destination transcription (either the orthography or the target phonemic transcription) is obtained by means of Dynamic Programming (DP) which is controlled by a predefined image set per unit that can appear in the source transcription and some easy to set control parameters. The image set of a source unit comprises all the units that can appear in a target transcription and that frequently co-occur with the source unit . Since it is generally known that certain graphemic patterns (e.g. “eau”, “ie”, “ij”, etc. in Dutch) often give rise to one sound, the sound-to-letter alignment can align a sound to sequences of up to four graphemes. Figure 4.3 shows a multi-character pattern “ssch” which is lined up with the source phoneme /s/. Since the image sets mostly represent domain independent knowledge, good baseline sets for a certain language can be constructed once, and later be reused for different domains. The user then has the opportunity to update the files manually on the basis of statistical information (most frequently observed sound-to-sound and sound-to-letter substitutions, number of deletions, insertions and substitutions within and outside the image sets) and to repeat the alignments with these new files.

4.3.2.2 Transformation Retrieval

In a second stage, the outputs of the aligner are analyzed in order to identify the (focus,output) transformations that can explain a large part of the observed discrepancies between the source transcriptions and the corresponding target transcriptions. Since stress markers are always lined up with stress markers (cf. previous section), and since every syllable is presumed to have a stress level of 0 (no stress), 1 (secondary stress) or 2 (primary stress), the stress transformations are restricted to stress substitutions. All of the six possible substitutions that occur frequently enough are retained as candidate stress transformations. The candidate phonemic transformations are retrieved from the computed alignments after removal of the stress markers. That retrieval process is governed by the following principles:

1. Consecutive source phonemes that differ from their corresponding target phonemes are kept together to form a single focus,
2. This agglomeration process is not interrupted by the appearance of a matching boundary pair (as we also want to model cross-syllable phenomena),

Fig. 4.4 Candidate transformations that can be retrieved from the alignment of Fig. 4.3

3. A focus may comprise a boundary symbol, but it cannot start/end with such a symbol (as we only attempt to learn boundary displacement rules, no boundary deletion or insertion rules),
4. (Focus,output) pairs are not retained if the lengths of focus and output are too unbalanced (a ratio >3), or if they imply the deletion/insertion of three or more consecutive phonemes,
5. (Focus,output) pairs not passing the unbalance test are split into two shorter candidate transformations whenever possible.

Once all utterances are processed, the set of discovered transformations is pruned on the basis of the phoneme discrepancy counts associated with these transformations. The phoneme discrepancy count expresses how many source phonemes would become equal to their corresponding target phoneme if the transformation were applied at the places where it helps (and not at any other place). Figure 4.4 shows one stress transformation (from primary to no stress) and three phonemic transformations (/I/,/i/), (/f/,/v/) and (/E n/,/@ m/) that comply with the five mentioned principles and that emerge from the alignment of Fig. 4.3.

4.3.2.3 Example Generation

Once the relevant transformation list is available, the focuses appearing in that list are used to segment the source transformation of each training object. The segmentation is performed by means of a stochastic automaton. This automaton represents a unigram model that comprises a set of phoneme consuming branches. Each branch corresponds to a single or multi-state focus model containing states to consume the subsequent phonemic symbols of the focus it represents. One additional branch represents a single-state garbage model that can consume any phonemic unit. Transition probabilities are automatically set so that a one-symbol focus will be preferred over the garbage model and a multi-state focus model will be preferred over a sequence of single state focus models. Once the segmentation of a source transcription is available, a training example will be generated for each focus segment encountered in that transcription. Recalling that we want to learn three types of correction rules: (1) stress substitution rules (SS-rules), (2) phoneme substitution and deletion rules (PSD-rules) and (3) phoneme insertion rules (PI-rules), we will also have to generate three types of examples. Each example consists of a rule input, a rule output and a set of features describing the linguistic context in which the rule input occurs.

4.3.2.4 Rule Induction

From the training examples, the system finally learns a decision tree for each focus appearing in the transformation list. The stochastic transformation rules are attached to each of the leaf nodes of such a tree. The identity rule (do not perform any transformation) is one of the stochastic rules in each leaf node. The collection of all learned decision trees constitutes the actual P2P converter. The decision trees are grown incrementally by selecting at any time the best node split one can make on the basis of a list of yes/no-questions concerning the transformation context and a node splitting evaluation criterion. The node splitting evaluation criterion is entropy loss. Since it has been shown that more robust trees can be learned if asking questions about whether a feature belongs to particular value class are allowed, we have accommodated the facility to specify such value classes for the different feature types that appear in the linguistic description of the training examples.

4.3.3 The Actual p2p Conversion

If the orthography is involved in the description of the correction rules, the p2p converter starts with performing an alignment between the initial phonemic transcription and the orthography.

The next step is to examine each syllable of the initial transcription and to apply the stress mark modification rules if the conditions are met.

The third step consists of a segmentation of the initial phonemic transcription into modifiable patterns and non-modifiable units (by means of the segmentation system that was applied before during rule induction). Once the segmentation is available, the pronunciation variant generator will try PI rules at the start of each non-empty segment and PSD rules at the start of each modifiable segment. If at a certain point one or more rules can be applied, different variants (including the one in which the input pattern is preserved) can be generated at the corresponding point in already created partial variants [7]. The output of the pronunciation variant generator is a tree shaped network representing different phonemic transcriptions with different attached probabilities. The p2p converter will select the transcription with the highest probability as the final phonemic transcription. Obviously, one can expect that in a number of cases this transcription will be identical to the initial transcription.

4.3.4 The Transcription Tool

In their simplest operation mode the AUTONOMATA transcription tools aim at providing phonetic transcriptions for a list of orthographic items, either words or sentences. The transcriptions are either generated by an embedded **standard g2p converter** of Nuance (see below), or by a tandem system also comprising

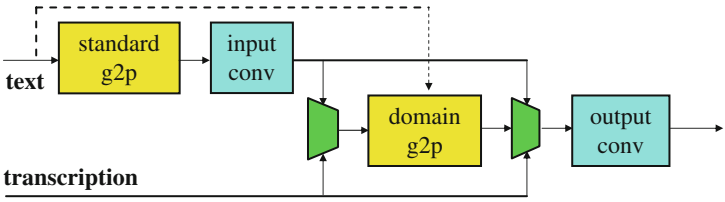


Fig. 4.5 Text-to-phoneme conversion in the autonomata transcription tools

a **domain specific phoneme-to-phoneme (p2p) converter** which tries to correct some of the mistakes made by the standard g2p converter and which also has access to the orthography. A third possibility is to select an already available phonetic transcription from the input file and to supply that to the p2p converter or to the **output conversion** block. In order to allow for a flexible use of different phonemic alphabets like CGN, LH+ (generated by the Nuance g2p converters) and YAPA (used in the HMM75 speech recognition engine), the transcription tools implement the transcription process depicted in Fig. 4.5.

The g2p always produces a transcription in terms of LH+ phonemes. The p2p converter can work directly on the g2p output or on a transformed version of it (e.g. transformed to CGN) obtained by an **input conversion** block. In the first case one does not have to specify any input conversion, in the other case one must specify one as explained below. The transcription tool can either select the transformed g2p transcription, the dedicated p2p output or a selected transcription from the input file. If needed, it can perform an additional conversion of this transcription, for instance, if the output transcriptions must be used in combination with a speech recognizer that is working with yet another phonemic alphabet.

In summary one can discern three phonetic alphabets: the **g2p-alphabet** (always LH+), the **p2p-alphabet** (LH+ or anything else being defined by the input conversion) and the **output-alphabet** (the p2p-alphabet or anything else being defined by the output conversion). In the simplest case all these alphabets are the same (LH+).

Since the p2p converters can operate on the output of the general-purpose g2p converter as well as on any automatic transcription that is already available in the input lexicon, it is easy to implement a cascade of p2p converters (let the first one operate on the g2p-output, the second one on the automatic transcription that was produced by the first p2p converter, etc.)

The transcription tool box is available via the Dutch HLT-agency.[10]

[10]See http://www.tst-centrale.org/nl/producten/tools/autonomata-transcriptietoolset/8-34

4.4 The Autonomata P2P Converters

The transcription tool comes with a number of p2p converters, yielding output in the CGN alphabet, for converting Dutch and Flemish person and place names:

- GeoNames_DUN: to use for Dutch geographical names in combination with DUN version of the Nuance g2p.
- GeoNames_DUB: to use for Flemish geographical names in combination with DUB version of the Nuance g2p.
- PersonNames_DUN: to use for Dutch person names in combination with DUN version of the Nuance g2p.
- PersonNames_DUB: to use for Flemish person names in combination with DUB version of the Nuance g2p.

Furthermore there are p2p converters for Points of Interest (POIs) developed in the Autonomata Too project:

- DUT_POI: to use in combination with DUN version of the Nuance g2p.
- ENG_POI: to use in combination with ENG version of the Nuance g2p.
- FRA_POI: to use in combination with FRF version of the Nuance g2p.

4.5 The Autonomata TOO POI Corpus

The Autonomata POI-corpus[11] was intended as an evaluation corpus for testing p2p converters developed for the transcription of Points of Interest (POIs) such as restaurants, hotels and rental companies. Such names often contain parts with an archaic or otherwise non-standard spelling as well as parts exhibiting a high degree of foreign influence.

4.5.1 Speakers

The corpus contains recordings of native speakers of Dutch, English, French, Turkish and Moroccan. The Dutch group consists of speakers from The Netherlands and Flanders. The English group contains speakers from the United States, Canada, Australia, Great Britain and Hong Kong. The other three groups consist of French, Turkish and Moroccan people, respectively. Table 4.3 contains the number of speakers in each group. Native speakers of Dutch will be referred to as Dutch speakers, speakers of foreign languages as foreign speakers. For both groups, this is a reference to native language, not to nationality.
Gender Speakers are equally distributed over age: 40 male and 40 female.

[11] This section is largely based on the corpus documentation written by Marijn Schraagen.

Table 4.3 Speaker distribution in the Autonomata TOO POI corpus

Mother tongue	Number of speakers
Dutch (Netherlands)	20
Dutch (Flanders)	20
English	10
French	10
Turkish	10
Moroccan	10
Total	80

Age All speakers are adults (above 18 years of age). Two categories are defined: younger than 40 years and 40 years or older. The Dutch speakers of each region (Netherlands and Flanders) are equally spread among these two groups. For foreign speakers, age has not been a strict criterion due to practical reasons.

Dialect region The dialect region is defined as in the ASNC and is only applicable to Dutch speakers

Education Education level is divided in two categories: high and low. The first category contains colleges (university and Dutch HBO schools), the second category contains all other levels of education. This variable has no strict distribution.

Home language The language spoken in informal situations is defined for Dutch speakers only. We distinguish three categories: standard Dutch, dialect, or a combination of standard Dutch and dialect. The assessment of this variable is left to the speaker, no criteria are defined for borders between the categories.

Number of years in Dutch language area and language proficiency For foreign speakers, the number of years they have lived in the Dutch language area is recorded. Besides this, we have asked all foreign speakers whether they have attended a language course for Dutch. If available, we have recorded the CEF level (Common European Framework for language proficiency). If the CEF level was not known by the speaker, we have indicated whether or not a language course was attended.

Foreign language proficiency All speakers were asked what languages they speak, and how proficient they are in every language: basic, intermediate, or fluent. The assessment of level is left to the speakers.

4.5.2 Recording Material

The POI list is designed in order to present 200 POI's to each speaker. The POI's are Hotel-Motel-Camp site and Café-Restaurant-Nightlife names that have been selected from the TeleAtlas POI database of real Points-of-Interest in The Netherlands and Belgium. The POI names were selected according to language. The list contains Dutch, English and French names, and names in a combination of either Dutch and English or Dutch and French.

Table 4.4 Reading lists with number of prompted items per speaker group

Speaker group	Number of speakers	Number of POI names		
		DU $_{50 = 30\,DU + 10\,(DU+EN) + 10\,(DU+FR)}$	EN	FR
Dutch group 1	10	50 (A)	75	75
Dutch group 2	10	50 (B)	75	75
Dutch group 3	10	50 (C)	75	75
Dutch group 4	10	50 (D)	75	75
Foreign	40	50 (A)+50 (B) + 50 (C) + 50 (D)	0	0

All foreign speakers read the same list, containing Dutch names and combination names. The focus of the Autonomata TOO research project did not require to present English or French names to foreign speakers.

Dutch speakers read one of four lists. Every list contains a unique selection of English and French names. Besides this, all Dutch lists contained a quarter of the Dutch names and the combination names from the foreign list. Table 4.4 shows the POI list construction with the number of names in each category. The colors and characters A–D indicate the list the names belong to.

Following this division, every Dutch name (including combination names) is spoken by 50 different speakers (all foreign speakers and 10 out of 40 Dutch speakers). Every French and English name is spoken by ten different speakers. The total list contains 800 names, of which 200 Dutch (120 Dutch only and 80 combination names with English or French), 300 English and 300 French names.

4.5.3 Recording Equipment and Procedure

The recordings were made using a software tool by Nuance, specially developed for the Autonomata TOO project and built as a GUI around the Nuance VoCon 3200 speech recognition engine, version 3.0F3. The speech recognition engine used a Dutch grapheme to phoneme converter from Nuance, and as a lexicon the full TeleAtlas POI set for The Netherlands and Belgium. The recognition engine used a baseline system of Dutch g2p transcriptions and Dutch monolingual acoustic models.

The recordings were made on a laptop with a USB headset microphone. Digital recordings were stored on the laptop hard disk in 16 bit linear PCM (wav-format). The sampling frequency is 16 kHz. We used a unidirectional Electret condensor microphone with a frequency range of 40–16 kHz.

The speaker was in control of the application. A POI name was shown on the screen, and the speaker started recording this name by pressing a button. Speech recognition was performed immediately on starting the recording, and the recognition result was shown to the speaker. The system checked whether the POI name was recognized correctly. On successful recognition, the system proceeded

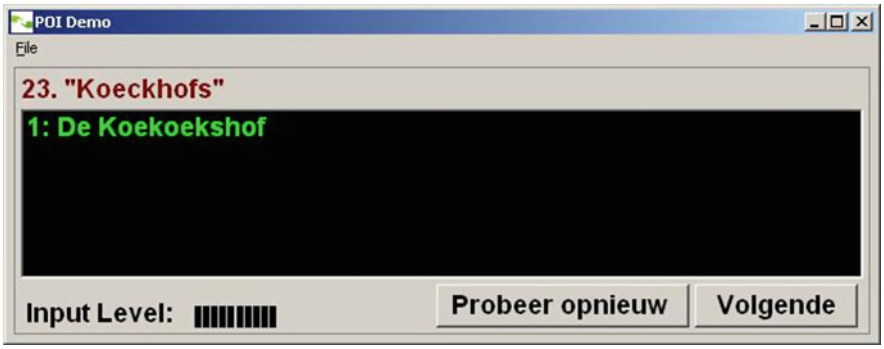


Fig. 4.6 Screenshot of the Autonomata POI database recording tool

to the next POI name. On failed recognition, the speaker was presented with a possibility to do an additional recording. The recognition result for the new attempt was again presented to the user. This process repeated itself until either the POI name was either recognized correctly or the user decided to proceed to the next item. The speaker could choose how many times s/he wanted to repeat a failed utterance, with a minimum of one repetition. This was to create a situation which is similar to using a real application in which a speaker will re-try and adopt after a misrecognized utterance [3]. All utterances were kept and stored on hard disk.

The screenshot in Fig. 4.6 illustrates the recording tool. An example is shown where the recognition failed.

The researcher was sitting next to the test subject during the entire experiment, to assist in using the recording tool and to control the process of repeating utterances. Any instruction during the experiment was aimed to improve the quality of the recording (such as preventing incomplete recordings or deliberately incorrect pronunciations), and to prevent useless repetitions (for example repeating an utterance more than three times in exactly the same way). The researcher did not answer questions regarding the correct pronunciation of an item. Before starting the real recording sessions, a number of ten test recordings was performed to let the user get acquainted to the recording tool works and to monitor the quality of the recordings. After the recording session, all recordings were checked. Incomplete recordings or recordings containing a severely mixed up reading were deleted.

The recordings were performed in sound-proof studios in Utrecht and Ghent. If test subjects were unable or unwilling to come to the record studio, the recording was performed on location. In this case, we have tried to minimize any influence from outside noise and reverberation.

4.5.4 Annotations

Each name token comes with an orthographical representation and an auditorily verified phonemic transcription (containing LH+ phonemes, word and syllable boundaries and primary stress markers). The latter were made in Praat in very much the same way as in the ASNC. The transcription protocol that was used is distributed together with the corpus.

4.5.5 Corpus Distribution

The corpus is 1.7 GB large and is distributed by the Dutch HLT-agency (TST-centrale).[12] The corpus documentation is very much similar to the one of the ASNC, but the POI-corpus also contains the ASR recognition results obtained with the Nuance VoCon 3200 recognition engine (version 3.0F1) during the recording process.

References

1. Réveil, B., Martens, J.-P., Van den Heuvel, H.: Improving proper name recognition by means of automatically learned pronunciation variants. Speech Commun. **54**(3), 321–340 (2012)
2. Schraagen, M., Bloothooft, G.: Evaluating repetitions, or how to improve your multilingual ASR system by doing nothing. Proceedings LREC2010, Malta (2010)
3. Van den Heuvel, H., Martens, J.-P., Konings, N.: G2P conversion of names. What can we do (better)? Proceedings Interspeech, Antwerp, pp. 1773–1776 (2007)
4. Van den Heuvel, H., Martens, J.-P., D'hoore, B., D'hanens, K., Konings, N.: The Autonomata Spoken Name Corpus. Design, recording, transcription and distribution of the corpus. Proceedings LREC 2008, Marrakech (2008)
5. Van den Heuvel, H., Martens, J.-P., Konings, N.: Fast and easy development of pronunciation lexicons for names. In: Proceedings LangTech 2008, Rome (2008)
6. van den Heuvel, H., Réveil, B., Martens, J.-P., D'hoore, B.: Pronunciation-based ASR for names. In: Proceedings Interspeech2009, Brighton (2009)
7. Yang, Q., Martens, J.-P., Konings, N., Van den Heuvel, H.: Development of a phoneme-to-phoneme (p2p) converter to improve the grapheme-to-phoneme (g2p) conversion of names. In: Proceedings LREC, Genua, pp. 287–292 (2006)

[12] See: http://www.tst-centrale.org/nl/producten/corpora/autonomata-poi-corpus/6-70

Chapter 5
STEVIN Can Praat

David Weenink

5.1 Introduction

Appropriate tools are indispensable for the scientist to perform his/her work. This holds true for speech science as well. Many tools exist in this field like, for example, Audacity [1], CSL [4], the MATLAB Signal Processing Toolbox [11] or Wavesurfer [14], to name a few. The Praat program [2] is an extensive application for language, music and speech research that is used by many scientists and students around the globe. A conservative guess from our website download statistics would indicate at least 20,000 users world-wide. Some characteristics that explain its success right from the beginning, are the wide range of features, the user-friendliness and the scriptability, i.e. the possibility to create ones own processing for a series of inputs. The other aspect that adds to the enthusiastic and widespread use is the careful support available. This encompasses user help on diverse levels online, quick response to any questions by email, immediate handling of incidents and solving of problems, and last but not least, an infrastructure for user groups. The knowledge that the Praat program entails, is in this means passed on to many colleagues and students. Also, users have a way to relate to one another and share their insights with regard to the possibilities the Praat program offers. The Praat software is freely available for most current computer platforms like Linux, Windows and Macintosh; it is not available on mobile devices. The manuals, FAQ and help menu are included in the package; the user group is available on the internet.[1] Despite the multitude of features already present in the application, some important functionality was

[1] http://groups.yahoo.com/group/praat-users

D. Weenink (✉)
University of Amsterdam and SpeechMinded, Amsterdam, The Netherlands
e-mail: David.Weenink@uva.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_5, © The Author(s) 2013

still missing. We have proposed to develop a number of improvements and added functionality that now has become freely available for speech scientists via the Praat program. This project matched the STEVIN objectives since it delivers important tools to all speech scientists who need state of the art technology to tackle the newest ideas and the largest datasets. The improvements that we have added to the Praat program are the following:

- KlattGrid: an acoustic synthesiser modeled after the Klatt synthesiser.
- VowelEditor: a sound-follows-mouse type editor by which vowel-like sounds can be generated from mouse gestures in the two dimensional formant plane.
- Robust formant frequency analysis.
- Availability of the mathematical functions from the GNU Scientific Library.
- Search and replace with regular expressions.
- Software band filter analysis.

In the rest of this chapter we will discuss these additions in somewhat more detail.[2]

5.2 The KlattGrid Acoustic Synthesiser

Although current speech synthesis is more oriented towards unit synthesis there is still need for a formant based synthesiser. A formant based speech synthesiser is a fundamental tool for those fields of speech research where detailed control of speech parameters is essential. For example, research on adult learner's vowel contrast in second language acquisition may require tight control over speech stimuli parameters while this also holds true for the investigation of vowel categorisation development of infants [6]. For the synthesis of different voices and voice characteristics and to model emotive speech formant based synthesis systems are still in use [12].

A very well known and widely used formant based speech synthesiser is the Klatt synthesiser [7,8]. One reason for its popularity is that the FORTRAN reference code was freely available as well as several C language implementations. In Fig. 5.1 we show a schematic diagram of this synthesiser with the vocal tract section realised with filters in cascade. Since a KlattGrid is based on the same design this is also the diagram of a KlattGrid. The synthesiser essentially consists of four parts:

1. The *phonation part* generates voicing as well as aspiration. It is represented by the top left dotted box labeled with the number 1 in its top right corner.
2. The *coupling part* models coupling between the phonation part and the next part, the vocal tract. In the figure it is indicated by the dotted box labeled with the number 2.
3. The *vocal tract part* filters the sound generated by the phonation part. The top right dotted box labeled 3 shows this part as a cascade of formant and

[2]The sections on the KlattGrid is a modified version of the [15] article.

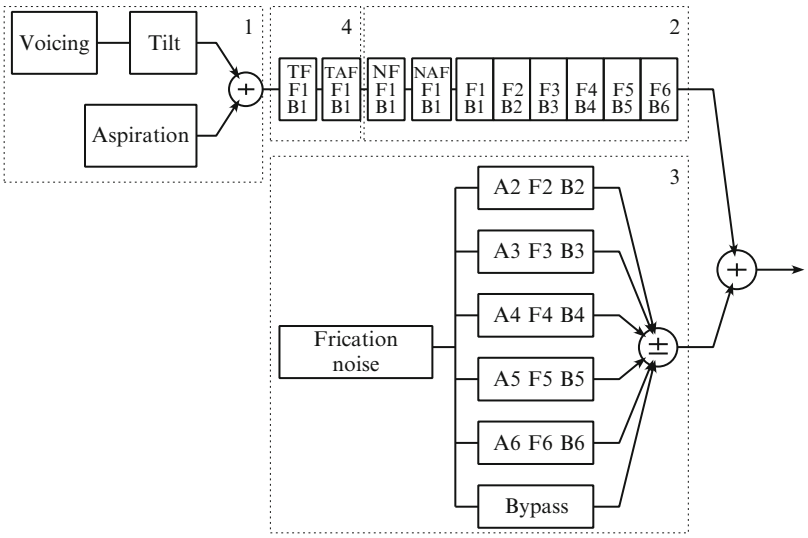


Fig. 5.1 Schematic diagram of a KlattGrid/Klatt synthesiser with the vocal tract part realised as filters in cascade

antiformant filters. The vocal tract part can also be modeled with formant filters that are in parallel instead of in cascade.

4. The *frication part* generates frication noise and is represented by the dotted box labeled 4.

A number of implementations of the Klatt synthesiser exist nowadays. However, they all show some of the limitations of the original design that originates from times that computer memory and processing power were relatively scarce. Necessarily, compromises had to be made at that time in order to achieve reasonable performance.

We present the KlattGrid speech synthesiser which is based on the original description of Klatt [7,8]. There are several new, not necessarily innovative, aspects in the KlattGrid in comparison with some other Klatt-type synthesisers.

- A Klatt synthesiser is frame-based, i.e. parameters are modeled to be constant during the interval of a frame, typically some 5 or 10 ms. As a consequence, instants of parameter change have to be synchronised on a frame basis. This poses some difficulty in modeling events where timing is important such as a rapidly changing amplitude for plosive bursts. We have removed this limitation by modeling *all* parameters in a KlattGrid as *tiers*. A tier represents a parameter contour as a function of time by (*time*, *value*) points. Parameter values at any time can be calculated from these time stamps by some kind of interpolation. For example, a formant frequency tier with two (*time*, *frequency*) points, namely 800 Hz at a time of 0.1 s and 300 Hz at 0.3 s, is to be interpreted as a formant frequency contour that is constant at 800 Hz for all times before 0.1 s, constant

at 300 Hz for all times after 0.3 s and linearly interpolated for all times between 0.1 and 0.3 s (i.e. 675 Hz at 0.15 s, 550 Hz at 0.2 s, and so on). By leaving the frame-based approach of previous synthesisers, all parameter timings become transparent and only moments of parameter change have to be specified.
- In a Klatt synthesiser one can normally define some six to eight oral formants and one nasal and one tracheal formant/antiformant pair. In a KlattGrid any number of oral formants, nasal formants and nasal antiformants, tracheal formants and tracheal antiformants are possible.
- In a Klatt synthesiser there is only one set of formant frequencies that has to be shared between the vocal tract part and the frication part. In a KlattGrid the formant frequencies in the frication part and the vocal tract part have been completely decoupled from one another.
- In the Klatt synthesiser the glottal flow function has to be specified beforehand. A KlattGrid allows varying the form of the glottal flow function as a function of times.
- In the Klatt synthesiser only the frequency and bandwidth of the first formant can be modified during the open phase. In a KlattGrid there is no limit to the number of formants and bandwidths that can be modified during the open phase of the glottis.
- In Klatt's synthesiser all amplitude parameters have been quantised to 1 dB levels beforehand. In a KlattGrid there is no such quantisation. All amplitudes are represented according to the exact specifications. Quantisation only takes place on the final samples of a sound when it has to be played or saved to disk (playing with 16-bit precision, for example). Of course sampling frequencies can be chosen freely.
- A KlattGrid is fully integrated into the speech analysis program Praat [2]. This makes the synthesiser available on the major desktop operating systems of today: Linux, Windows and Mac OS X. At the same time all scripting, visualisations and analysis methods of the Praat program become directly available for the synthesised sounds.

More details on the KlattGrid can be found in the following sections which will describe the four parts of the synthesiser in more detail. This description will be a summary of the synthesiser parameters and how they were implemented.

5.2.1 The Phonation Part

The phonation part serves two functions:

1. It generates voicing. Part of this voicing are timings for the glottal cycle. The part responsible for these timings is shown by the box labeled "Voicing" in Fig. 5.1. The start and end times of the open phase of the glottis serve to:
    - Generate glottal flow during the open phase of the glottis.
    - Generate breathiness, i.e. noise that occurs only during the open phase of the glottis.

- Calculate when formant frequencies and bandwidths change during the open phase (if formant change information is present in the coupling part).

2. It generates aspiration. This part is indicated by the box labeled "Aspiration" in Fig. 5.1. In contrast with breathiness, aspiration may take place independently of any glottal timing.

The phonation parameter tiers do not all independently modify the glottal flow function. Some of the parameters involved have similar spectral effects, however, in this article we do not go into these details too much and only briefly summarise a tier's function in the phonation part. For an extensive overview of the effects of several glottal flow parameters on the source spectrum see for example the article of Doval et al. [5]. The following 11 tiers form the phonation part:

Pitch tier. For voiced sounds the pitch tier models the fundamental frequency as a function of time. Pitch equals the number of glottal opening/closing cycles per unit of time. In the absence of flutter and double pulsing, the pitch tier is the only determiner for the instants of glottal closure. Currently pitch interpolation happens on a linear frequency scale but other interpolation, for example on a log scale, can be added easily.

Voicing amplitude tier. The voicing amplitude regulates the maximum amplitude of the glottal flow in dB. A flow with amplitude 1 corresponds to approximately 94 dB. To produce a voiced sound it is essential that this tier is not empty.

Flutter tier. Flutter models a kind of "random" variation of the pitch and it is input as a number from zero to one. This random variation can be introduced to avoid the mechanical monotonic sound whenever the pitch remains constant during a longer time interval. The fundamental frequency is modified by a flutter component according to the following semi-periodic function that we adapted from [7]: $F_0'(t) = 0.01 \cdot \text{flutter} \cdot F_0 \cdot (\sin(2\pi 12.7t) + \sin(2\pi 7.1t) + \sin(2\pi 4.7t))$

Open phase tier. The open phase tier models the open phase of the glottis with a number between zero and one. The open phase is the fraction of one glottal period that the glottis is open. The open phase tier is an optional tier, i.e. if no points are defined then a sensible default for the open phase is taken (0.7). If the open phase becomes smaller, necessarily the high frequency content of the source spectrum will increase.

Power1 and power2 tiers. These tiers model the form of the glottal flow function during the open phase of the glottis as $\text{flow}(t) = t^{\text{power1}} - t^{\text{power2}}$, where $0 \leq t \leq 1$ is the relative time that runs from the start to the end of the open phase. For the modelation of a proper vocal tract flow it is essential that the value of power2 is always larger than the value of power1. If these tiers have no values specified by the user, default values power1 = 3 and power2 = 4 are used. Figure 5.2 will show the effect of the values in these tiers on the form of the flow and its derivative. As power2 mainly influence the falling part of the flow function, we see that the higher the value of this parameter, the faster the flow function reaches zero, i.e. the shorter the closing time of the glottis would be and, consequently, the more high frequency content the glottal spectrum will have.

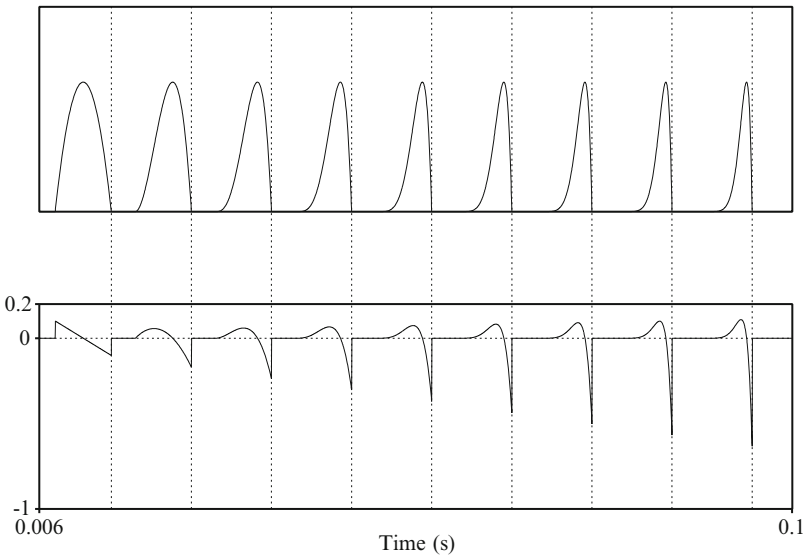


Fig. 5.2 On *top*, nine glottal pulses flow$(t) = t^{\text{power1}} - t^{\text{power2}}$, each one synthesised with a different (power1, power2) combination. Power1 increases linearly from 1, and always power2 = power1 + 1. Consequently, the first pulse on the *left* has power1 = 1 and power2 = 2, while the last one on the *right* has power1 = 9 and power2 = 10. The *bottom panel* on the *left* shows the derivatives of these flow functions. Moments of glottal closure have been indicated with a *vertical dotted line*. The open phase was held at the constant value of 0.7, the pitch was fixed at 100 Hz and the amplitude of voicing was fixed at 90 dB

Collision phase tier. The collision phase parameter models the last part of the flow function with an exponential decay function instead of a polynomial one. A value of 0.04, for example, means that the amplitude will decay by a factor of $e \approx 2.7183$ every 4 % of a period. The introduction of a collision phase will reduce the high frequency content in the glottal spectrum because of the smoother transition towards the closure.

Spectral tilt tier. Spectral tilt represents the extra number of dB's the voicing spectrum should be tilted down at 3,000 Hz [7]. This parameter is necessary to model "corner rounding", i.e. when glottal closure is non simultaneous along the length of the vocal folds. If no points are defined in this tier, spectral tilt defaults to 0 dB and no spectral modifications are made.

Aspiration amplitude tier. The aspiration amplitude tier models the (maximum) amplitude of noise generated at the glottis. The aspiration noise amplitude is, like the voicing amplitudes, specified in dB. This noise is independent of glottal timings and is generated from random uniform noise which is filtered by a very soft low-pass filter.

Breathiness amplitude tier. The breathiness amplitude tier models the maximum noise amplitude during the open phase of the glottis. The amplitude of the breathiness noise, which is plain random uniform noise, is modulated by the glottal flow. It is specified in dB.

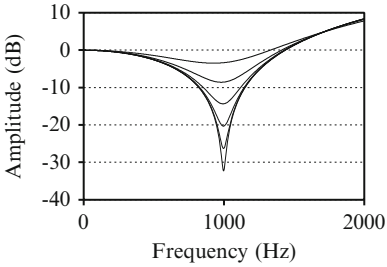


Fig. 5.3 Example of frequency responses of formant/antiformant pairs

Double pulsing tier. The double pulsing tier models diplophonia (by a number from zero to one). Whenever this parameter is greater than zero, alternate pulses are modified. A pulse is modified with this single parameter tier in two ways: it is delayed in time and its amplitude is attenuated. If the double pulsing value is maximum (=1), the time of closure of the first peak coincides with the opening time of the second one (but its amplitude will be zero).

5.2.2 The Vocal Tract Part

The sound generated by the phonation part of a KlattGrid may be modified by the filters of the vocal tract part. These filters are the oral formant filters, nasal formant filters and nasal antiformant filters. A formant filter boosts frequencies and an antiformant filter attenuates frequencies in a certain frequency region. For speech synthesis the vocal tract formant filters can be used in cascade or in parallel. Default these filters are used in cascade as is shown in Fig. 5.1 in the part numbered *3*, unless otherwise specified by the user. Each formant filter is governed by two tiers: a formant *frequency* tier and a formant *bandwidth* tier. In case of parallel synthesis an additional formant *amplitude* tier must be specified.

Formant filters are implemented in the standard way as second order recursive digital filters of the form $y_n = ax_n + by_{n-1} + cy_{n-2}$ as described in [7] (x_i represents input and y_j output). These filters are also called digital resonators. The coefficients b and c at any time instant n can be calculated from the formant frequency and bandwidth values of the corresponding tiers. The a parameter is only a scaling factor and is chosen as $a = 1 - b - c$; this makes the frequency response equal to 1 at 0 frequency. Antiformants are second order filters of the form $y_n = a'x_n + b'x_{n-1} + c'x_{n-2}$. The coefficients a', b' and c' are also determined as described in [7]. When formant filters are used in cascade all formant filters start with the same value at 0 Hz. If used in parallel this is not the case anymore since each formant's amplitude must be specified on a different tier.

As an example we show in Fig. 5.3 the frequency responses of formant/antiformant pairs where both formant and antiformant have the same "formant" frequency, namely 1,000 Hz, but different bandwidths. The bandwidth of the antiformant filter

was fixed at 25 Hz but the bandwidth of the formant filter doubles at each step. From top to bottom it starts at 50 Hz and then moves to 100, 200, 400 and 800 Hz values. A perfect spectral "dip" results without hardly any side-effect on the spectral amplitude. This shows that a combination of a formant and antiformant at the same frequency can model a spectral dip: the formant compensates for the effect on the slope of the spectrum by the antiformant. Best spectral dips are obtained when the formant bandwidth is approximately 500 Hz. For larger bandwidths the dip will not become any deeper, the flatness of the spectrum will disappear and especially the higher frequencies will be amplified substantially.

5.2.3 The Coupling Part

The coupling part of a KlattGrid models the interactions between the phonation part, i.e. the glottis, and the vocal tract. Coupling is only partly shown in Fig. 5.1, only the tracheal formants and antiformants are shown. We have displayed them in front of the vocal tract part after the phonation part because tracheal formants and antiformants are implemented as if they filter the phonation source signal.

Besides the tracheal system with its formants and antiformants the coupling part also models the change of formant frequencies and bandwidths during the open phase of the glottis. With a so-called *delta formant grid* we can specify the amount of change of any formant and/or bandwidth during the open phase of the glottis. The values in the delta tiers will be added to the values of the corresponding formant tiers but *only during the open phase of the glottis*.

In Fig. 5.4 we show two examples where extreme coupling values have been used for a clear *visual* effect. In all panels the generated voiced sounds had a constant 100 Hz pitch, an constant open phase of 0.5 to make the duration of the open and closed phase equal, and only one formant. In the left part of the figure formant bandwidth is held constant at 50 Hz while formant frequency was modified during the open phase. The oral formant frequency was set to 500 Hz. By setting a delta formant point to a value of 500 Hz we accomplish that during the start of the open phase of the glottis, the formant frequency will increase by 500–1,000 Hz. At the end of the open phase it will then decrease to the original 500 Hz value of the formant tier. To avoid instantaneous changes we let the formant frequency increase and decrease with the delta value in a short interval that is one tenth of the duration of the open phase. In a future version of the synthesiser we hope to overcome this limitation [13]. The top display at the left shows the actual first formant frequency as a function of time during the first 0.03 s of the sound. This is exactly the duration of three pitch periods; the moments of glottal closure are indicated by dotted lines. The bottom left display shows the corresponding one-formant sound signal. The 100 Hz periodicity is visible as well as the formant frequency doubling in the second part of each period: we count almost two and a half periods of this formant in the first half of a period, the closed phase, and approximately five during the second half of a period, the open phase. At the right part of the same figure we show the effect of a

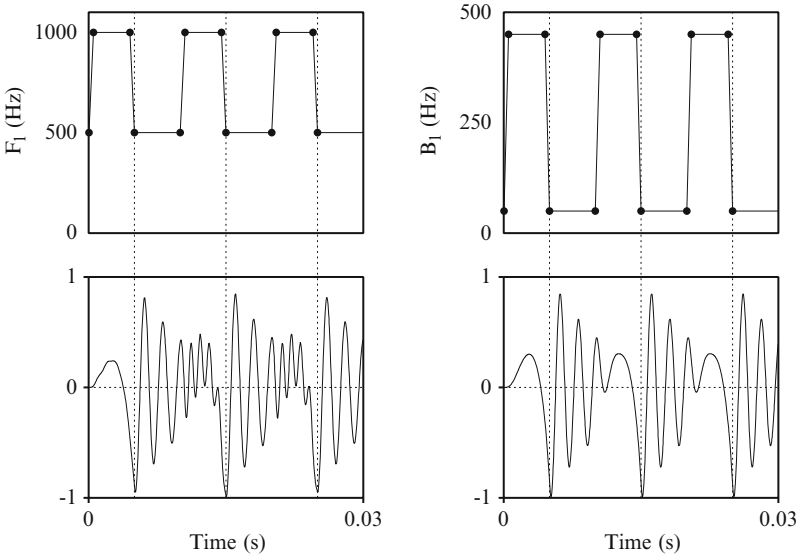


Fig. 5.4 Synthesised example of coupling between vocal tract and the glottis with some extreme formant and bandwidth coupling values. Formant and bandwidth tier at *top*, resulting sounds at *bottom*. Glottal closure times are indicated with *dotted lines*

bandwidth increase from 50 Hz during the closed phase to 450 Hz during the open phase for a one-formant vowel. As before the increase and decrease occur during the first and last one tenth of the open phase interval as is shown in the top right panel. The bottom right panel shows the corresponding synthesised sound.

5.2.4 The Frication Part

The frication part is an independent section in the synthesiser and it gives the opportunity to add the frication noise completely independent of the phonation and the vocal tract part. The frication sound is added to the output of the vocal tract part. A layout of the frication part is shown at the bottom of Fig. 5.1 in the dotted box labeled 4. The following tiers specify the frication sound:

Frication amplitude tier. This tier regulates the maximum amplitude of the noise source in dB before any filtering takes place. In Fig. 5.1 this part is represented by the rectangle labeled "Frication noise". This noise source is uniformly distributed random noise.

Formant frequency and bandwidth tiers. To shape the noise spectrum a number of parallel formant filters are available whose frequencies, bandwidths and amplitudes can be specified. In the figure we have limited the number of formants to five but in principle this number is not limited at all.

Formant amplitude tiers. Each formant is governed by a separate amplitude tier with values in dB. These formant amplitudes act like multipliers and may amplify or attenuate the formant filter input. For formant amplitudes 0 dB means an amplification of 1. Formants can be increased by giving positive dB values and decreased by giving negative values.

Bypass tier. The bypass tier regulates the amplitude of the noise that bypasses the formant filters. This noise is added straight from the noise source to the output of the formant filters. The amplitude is in dB's, where 0 dB means a multiplier of 1.

5.2.5 A KlattGrid Scripting Example

As the preceding sections have shown, the KlattGrid has a large number of parameters. It is difficult to get to grips with the many ways of changing a synthesiser's sound output. To facilitate experimenting with parameter settings, the user interface has been designed to make it easy to selectively include or exclude, in the final sound generation process, some of the parameter tiers that you have given values. For example, if the breathiness amplitude has been defined, hearing the resulting sound with or without breathiness is simply achieved by a selection or deselection of the breathiness tier option in the form that regulates this special playing mode of the KlattGrid synthesiser. The same holds true for the phonation part of the synthesiser whose sound output can be generated separately with some of its parameter tiers selectively turned on or off.

As an example of the synthesisers interface we show a simple example script to synthesise a diphthong. This script can be run in Praat's script editor. The first line of the script creates a new KlattGrid, named "kg", with start and end times of 0 and 0.3 s, respectively. The rest of the parameters on this line specify the number of filters to be used in the vocal tract part, the coupling part and the frication part and are especially important for now (additional filters can always be added to a KlattGrid afterwards).

The second line defines a pitch point of 120 Hz at time 0.1 s. The next line defines a voicing amplitude of 90 dB at time 0.1 s. Because we keep voicing and pitch constant in this example the exact times for these points are not important, as long as they are within the domain on which the kg KlattGrid is defined. With the pitch and voicing amplitude defined, there is enough information in the KlattGrid to produce a sound and we can now Play the KlattGrid (line 4).[3] During 300 ms you will hear the sound as produced by the glottal source alone. This sound normally would be filtered by a vocal tract filter. But we have not defined the vocal tract filter yet (in this case the vocal tract part will not modify the phonation sound).

[3]Despite the fact that it will play correctly, you will receive a warning because not all parameters in the KlattGrid have been specified. For example, the oral formants have not been specified thus far.

In lines 5 and 6 we add a first oral formant with a frequency of 800 Hz at time 0.1 s, and a bandwidth of 50 Hz also at time 0.1 s. The next two lines add a second oral formant at 1,200 Hz with a bandwidth of 50 Hz. If you now play the KlattGrid (line 9), it will sound like the vowel /a/, with a constant pitch of 120 Hz. Lines 10 and 11 add some dynamics to this sound; the first and second formant frequency are set to the values 350 and 600 Hz of the vowel /u/; the bandwidths have not changed and stay constant with values that were defined in lines 6 and 8. In the interval between times 0.1 and 0.3 s, formant frequency values will be interpolated. The result will now sound approximately as an /au/ diphthong.

This script shows that with only a few commands we already may create interesting sounds.

```
Create KlattGrid... kg 0 0.3 6 0 0 0 0 0 0
Add pitch point... 0.1 120
Add voicing amplitude point... 0.1 90
Play
Add oral formant frequency point... 1 0.1 800
Add oral formant bandwidth point... 1 0.1 50
Add oral formant frequency point... 2 0.1 1200
Add oral formant bandwidth point... 2 0.1 50
Play
Add oral formant frequency point... 1 0.3 350
Add oral formant frequency point... 2 0.3 600
Play
```

5.3 Vowel Editor

Although we can perfectly specify sounds with the KlattGrid synthesiser this is not always the most convenient way. Especially for simple vowel-like sounds we need a more intuitive way to specify them. For didactic and demonstration purposes, a straightforward vowel generator of the type sound-follows-mouse has been implemented. We will now present some of its functionality without going into much detail.

By pressing the left mouse button and moving around the mouse pointer in the plane formed by the first two formants, a sound with the formant frequencies of this mouse pointer trajectory will be generated whenever the left mouse button is released. In Fig. 5.5 we show an example of a vowel editor window where the trajectory is indicated with a fat solid line. The main part of the vowel editor shows the plane formed by the first and second formant where the origin is in the upper right corner. The first formant frequency runs from top to bottom while the second formant frequency runs from right to left and the frequency scales are logarithmic, not linear. The dotted lines in the formant plane mark equidistant intervals. At the bottom of the window some characteristics of the generated trajectory are displayed

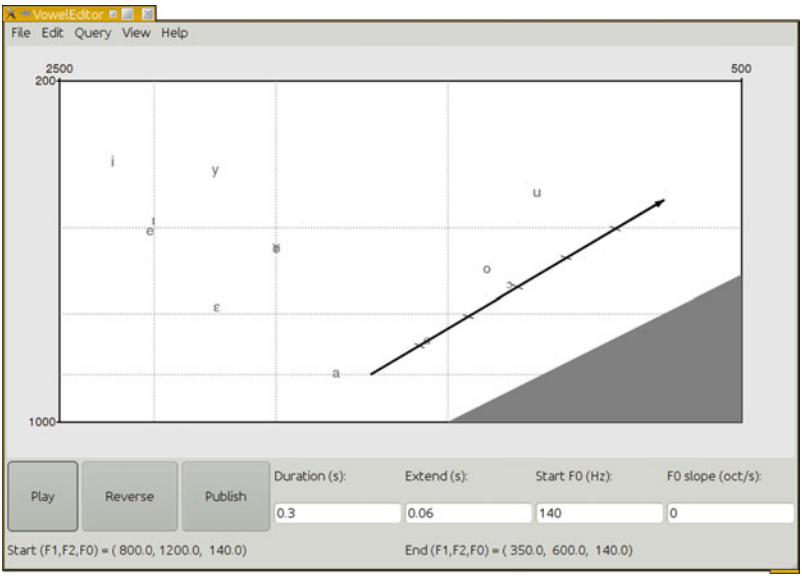


Fig. 5.5 The vowel editor's interface. The *fat line* shows a straight trajectory that starts with a first formant frequency of 800 Hz and the second formant at 1,200 Hz and that ends with a first formant of 350 Hz and the second at 600 Hz. The little bars on the trajectory mark 50 ms time intervals

such as the first two formant frequencies at the start and at the end points of the trajectory. Because the positions of the Dutch vowels in the formant plane are displayed too, we note that this trace corresponds to an /au/ type of diphthong sound. Hitting the big buttons labeled "Play" below the formant plane with the mouse generates a sound according to the trajectory specification and then plays this sound; for the displayed trajectory this will sound as /au/. Hitting the "Reverse" button will reverse the trajectory and then play it; for the displayed trajectory this will sound as /ua/. Apart from generating trajectories by moving the mouse around, trajectories can also be specified (and extended) from vowel editors menu options. For example, with the option "New trajectory . . ." from the Edit menu you can specify the first and second formant frequencies of the start and end points together with the duration of the trajectory. The start and end point will then be connected with a straight line. In fact the trajectory shown in Fig. 5.5 was specified as such. The start first and second formant frequencies and the end first and second formant frequencies had the same values as for the diphthong generated by the script in the previous section. More trajectory generating options exist but these will not be discussed here.

5.4 Robust Formant Frequency Analysis

Many phoneticians rely on formant frequency measurements for their investigations. The current formant frequency measurements in Praat are based on LPC-analysis performed by standard algorithms available in the literature [3, 10]. The formant

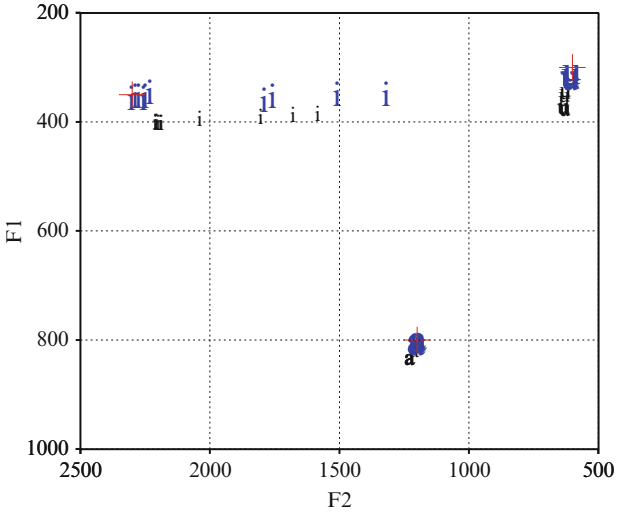


Fig. 5.6 F_1 versus F_2 measure with two different procedures on artificial vowels /u/, /i/ and /a/. The formant frequencies used for the vowel synthesis are indicated with crosses. The outcome of the standard Burg algorithm is indicated with the *smaller font* while the robust analysis is indicated with the larger sans serif font (*blue color*)

frequencies derived by these LPC-analyses algorithms can be improved upon by incorporating knowledge gained from robust statistics. The new formant analysis procedure first down-samples the sound to a frequency twice the highest formant frequency one is interested in. Default this means down-sampling to 10 kHz for male voices and down-sampling to 11 kHz for female voices. Next, LPC coefficients will be determined by the autocorrelation method. Finally, the prediction coefficients are refined by an iterative procedure which uses selective weighing of sample values, i.e. sample values with large prediction errors will be down-weighted as described in [9]. Our own tests indicate substantial improvements in formant frequency and bandwidth accuracy on artificially generated signals and significantly less variation in these measurements with analysis window position for sufficiently long windows that overlap several pitch periods. Of course, this does not immediately guarantee large measurement stability improvements for real speech signals, however, if we can improve significantly the precision of the formant frequency and band filter measurements on test signals, this mere fact should give us more confidence in the newer robust analysis method as compared to the standard methods. Although no known methods exist that always give the "correct" formant frequency values, making some of the current methods more robust with respect to analysis window position should have high priority. This is useful in all research where formant frequencies are used. Especially for high pitched voices this is potentially very interesting as it can be used in basic phonetic research where the development of vowel spaces for young children is the domain.

In Fig. 5.6 we show formant frequencies as measured on three synthetic vowels /u/, /i/ and /a/ by two different algorithms. The first is the Burg algorithm, which is

the standard in Praat for doing formant frequency analysis. The second one is our implementation of the robust LPC algorithm. The vowels were synthesised with an impulsive source of constant pitch. These three vowels are indicated with a cross in the figure. The third, fourth and fifth formant frequencies were set at values of 2,500, 3,500 and 4,500 Hz, respectively. In order to make the analysis more realistic, we varied the frequency interval for the LPC analysis from 4,000 to 5,500 Hz in ten steps. The figure therefore shows ten vowels symbols per analysis method. As we can see the outcome of the robust analysis lie closer to the target values for /u/ and /a/. Only for /i/ for the four lowest analysis frequency intervals, the values are off. However, the rest of the measurement values again lie closer. To show the possible improvement of the robust analysis more tests are necessary on which we will report elsewhere [16].

5.5 Availability of the Mathematical Functions in the GNU Scientific Library

The GNU Scientific Library (GSL) is a GPL licenced library with high quality basic elementary scientific functions.[4] It covers many aspects of mathematics such as special functions like the Bessel, gamma, Legendre and error functions, as well as permutations, linear algebra matrix operations and eigensystems. Statistical functions and distributions are also part of the library. Incorporation of this library therefore facilitates advanced functionality. The GSL library is in constant development. Praat's functionality was extended by making the complete library linkable with Praat. However, we never use the available GSL functions directly but only through a numerical interface. We have also setup an infrastructure for intensive testing of the used GSL functionality. Besides delivering information about valid input ranges of functions it gives information about numerical precision. This information can be used to test whether newer versions of the library comply with current precision constraints. The generation of intensive functional tests before actually writing the code is also heavily promoted by current software theory like Extreme Programming.[5]

5.6 Search and Replace with Regular Expressions

Before we completed this project, the search and replace functions that were available in Praat could only search and replace literal text. By implementing a regular expression engine[6] we have now substantially improved these possibilities by also

[4] http://www.gnu.org/software/gsl/

[5] http://www.extremeprogramming.org/rules/testfirst.html

[6] This regular expression engine is part of the GPL licenced programming editor *nedit*, available via http://www.nedit.org.

allowing regular expressions as patterns for search and replace. The new search and replace functions are available at all levels in Praat, in text-based data types used for annotation such as a TextGrid as well as for Praat's scripting language.

5.7 Software Band Filter Analysis

Software band filter analysis is a kind of spectral analysis and an essential part of any speech analysis program. In general band filter analysis by software proceeds by first dividing the sound in overlapping segments, making a spectrum of each segment and then binning the spectral values. The different band filter methods generally differ in how this binning is performed. For example, in Praat's implemented MelFilter object the binning is performed by overlapping triangular filters. On the mel frequency scale[7] these filters all have the same width. This type of band filter analysis will result in substantial data reduction as the hundreds of spectral values of each segment's spectrum are reduced to some 30 or less mel band filter values. The mel band filter representations can be graphically displayed or used for further processing to mel frequency cepstral coefficients. Further details can be found in the help available in the Praat program an in [16].

5.8 Conclusion

In the field of speech signal processing, Praat is the most widely used computer program. It is still gaining in acceptance. Keeping up-to-date with current techniques and user interface design is very important for all these enthusiastic current day users. Our STEVIN project has successfully implemented a number of useful additions to the Praat program that are being used by Praat's users all over the globe.

Acknowledgements I would like to thank prof. dr. Paul Boersma, project coordinator and release coordinator of the Praat program, for his enthusiasm and his constant emphasis on striving for the highest quality possible. Further thanks go to the other members of the research team prof. dr. Frans Hilgers, prof. dr. Vincent van Heuven and dr. Henk van den Heuvel for their support. Final thanks go to the reviewers for their valuable comments and suggestions.

[7] To go from frequency f in hertz to mel: $\text{mel}(f) = 2{,}595 \cdot \log(1 + f/700)$.

References

1. Audacity (Computer Program). http://audacity.sourceforge.net/
2. Boersma, P., Weenink, D.J.M.: Praat: doing phonetics by computer (computer program). http://www.praat.org/ (2012)
3. Childers, D.: Modern Spectrum Analysis. IEEE Press, New York (1978)
4. CSL Computerized Speech Lab (Computer Program). http://www.kayelemetrics.com/ (2012)
5. Doval, B., d'Alessandro, C., Henrich, N.: The spectrum of glottal flow models. Acta Acust. **92**, 1026–1046 (2006)
6. Escudero, P., Benders, T., Wanrooij, K.: Enhanced bimodal distributions facilitate the learning of second language vowels. J. Acoust. Soc. Am. Express Lett. **130**, 206–212 (2011)
7. Klatt, D.H.: Software for a cascade/parallel formant synthesizer. J. Acoust. Soc. Am. **67**, 971–995 (1980)
8. Klatt, D.H., Klatt, L.C.: Analysis, synthesis, and perception of voice quality variations among female and male talkers. J. Acoust. Soc. Am. **87**, 820–857 (1990)
9. Lee, C.H.: On robust linear prediction of speech. IEEE Trans. Acoust. Speech Signal Process. **36**, 642–649 (1988)
10. Markel, J.D., Gray Jr., A.H.: Linear Prediction of Speech. Springer Verlag, Berlin (1976)
11. MATLAB Signal Processing Toolbox (Computer Program). http://www.mathworks.es/products/signal/index.html (2012)
12. Ternström, S., Sundberg, J.: Formant-based synthesis of singing. In: Proceedings of Eurospeech 2, Antwerp, pp. 4013–4014 (2007)
13. Verhelst, W., Nilens, P.: A modified-superposition speech synthesizer and its application. In: Proceedings of ICASSP, Tokyo, pp. 2007–2010 (1986)
14. Wavesurfer (Computer Program). http://www.speech.kth.se/wavesurfer/ (2012)
15. Weenink, D.J.M.: The KlattGrid acoustic speech synthesizer. In: Proceedings Interspeech 2009, Brighton, pp. 2059–2062 (2009)
16. Weenink, D.J.M.: Speech signal processing by Praat. http://www.fon.hum.uva.nl/david/sspbook/sspbook.pdf (2012, To be published)

Chapter 6
SPRAAK: Speech Processing, Recognition and Automatic Annotation Kit

Patrick Wambacq, Kris Demuynck, and Dirk Van Compernolle

6.1 Introduction

Over the past years several users (in Belgium, the Netherlands and abroad) have adopted the ESAT speech recognition software package (developed for over 15 years at ESAT, K.U.Leuven, [5, 10]) as they found that it satisfied their research needs better than other available packages. However, typical of organically grown software, the learning effort was considerable and documentation lacking. The software needed a major overhaul and this is accomplished with support from the STEVIN programme and a partnership consisting of Katholieke Universiteit Leuven, Radboud University Nijmegen, Twente University and TNO. At the same time the main weaknesses were addressed and the code base was modernised. This effort also addressed the need of the research community for a Dutch speech recognition system that can be used by non-specialists. We also found this to be an ideal moment to open up the code to the community at large. It is now distributed as open source for academic usage and at moderate cost for commercial exploitation. The toolkit, its documentation and references can be found at http://www.spraak.org.

In this article details of the SPRAAK toolkit are given in several sections: Sect. 6.2 discusses the possible uses of the toolkit, Sect. 6.3 explains the features of the different components of the software, some benchmark results are given in Sect. 6.4, system requirements of the software are mentioned in Sect. 6.5, licensing and distribution is covered in Sect. 6.6, relation to other STEVIN projects in Sects. 6.7 and 6.8 addresses future work. Finally a conclusion is given in Sect. 6.9.

This article tries to give as complete information as possible about SPRAAK and is therefore targeted at several audiences:

P. Wambacq (✉) · K. Demuynck · D. Van Compernolle
Katholieke Universiteit Leuven, ESAT, Kasteelpark Arenberg 10 box 2441,
B3001 Heverlee, Belgium
e-mail: patrick.wambacq@esat.kuleuven.be; kris.demuynck@esat.kuleuven.be;
dirk.vancompernolle@esat.kuleuven.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*,
Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_6,

- The speech and language technology community at large. These readers will find most useful information in Sects. 6.2 (introduction and Sect. 6.2.1), 6.3.1, 6.4 and 6.7;
- Potential non-expert users, who could additionally be interested in the other parts of Sect. 6.2, and in Sects. 6.5, 6.6 and 6.8;
- Speech recognition researchers who are also served with the technical details in Sect. 6.3.

6.1.1 Alternatives to SPRAAK

When the SPRAAK project was conceived, a survey of open source alternatives was made. There exist other alternatives that are either commercial or limited to internal use. Commercial systems were not considered because of their price, issues with research licenses or unavailability of the source code. At the time of the start of the project the most important competitors were the following:

- The freely available version of the well-known HTK toolkit [14] which has a less capable decoder which is limited to bigram grammars and is not fit for real-time applications. The code also has a complex global structure which makes it difficult to change.
- The CMU Sphinx4 system [3] which does not have state-of-the-art performance. The choice to implement the system in Java provides excellent flexibility, but is detrimental to the memory footprint of the system and limits the decoding speed.
- Julius [16] which is a fairly successful Japanese recogniser but the documentation (at that time only in Japanese) and language barrier in general is however an important issue.

We believe that the above mentioned obstacles have not yet been removed, except that now English documentation is being written for Julius. Since the start of the SPRAAK project, another open source initiative has seen the light in 2009: the RWTH ASR toolkit [20]. We don't have any experience with this software but from the documentation available on the website, features seem to be comparable to ours (with some extras but also some shortcomings).

6.2 Intended Use Scenarios of the SPRAAK Toolkit

SPRAAK is a speech recognition toolkit intended for two distinct groups of users:

- SPRAAK is designed to be a flexible modular toolkit for speech recognition research, yet at the same time
- SPRAAK provides a state-of-the art speech recogniser suitable for speech application deployment in niche markets.

In order to address the different needs, functionalities in SPRAAK may be addressed by a set of different interfaces. Application developers need control over

the run-time engine and over the resources that the engine uses. They are served by the High Level API (Application Programmers Interface). Developing new resources may imply the usage of and minor adaptations to a number of standalone scripts that make use of the lower level API.

Speech researchers will be served by the scripting framework for training and evaluating new systems. Certain types of speech research (e.g. feature extraction) may involve only limited programming, while others (e.g. changes to the decoder) are always a major programming effort. Depending on the type of research they may need to dig deeper into the internals for which they will make use of the low level API.

A speech recognition system is a piece of complex software and although an inexperienced user does not have know its inner workings, some basic knowledge is required to be able to use it. To this end we have organised hands-on seminars, and this material was then later reused to produce new and better tutorials that are now part of the documentation. Given enough interest, more seminars can be organised in the future.

6.2.1 SPRAAK as a Research Toolkit

SPRAAK is a flexible modular toolkit for research into speech recognition algorithms, allowing researchers to focus on one particular aspect of speech recognition technology without needing to worry about the details of the other components. To address the needs of the research community SPRAAK provides:

- Plug and play replacement of the core components
- Flexible configuration of the core components
- Extensive libraries covering the common operations
- Examples on how to add functionality
- Well defined and implemented core components and interaction schemes
- Programmers and developers documentation
- A low level API
- A direct interface with the Python scripting language
- An interactive environment (Python) for speech research
- Scripts for batch processing

6.2.2 SPRAAK for Application Development

SPRAAK is also a state-of-the art recogniser with a programming interface that can be used by non-specialists with a minimum of programming requirements. At the same time SPRAAK allows ASR specialists to integrate and harness special functionality needed for niche market projects that are not served by the existing

off-the-shelf commercial packages. To address the needs of this part of the user base, SPRAAK provides:

- Users documentation;
- A high level API;
- A client-server model;
- Standard scripts for developing new resources such as acoustic models.
- Demo models can be obtained from the Dutch-Flemish HLT Agency; these can serve as starting points for a recogniser and to train own models. This includes models for Northern and Southern Dutch for both broadband and telephony speech and a set of resources (acoustic and language models, lexica, . . .);
- On request the developers can also provide a set of reference implementations or frameworks for different applications.

6.2.3 SPRAAK Applications

Although it is possible to build conventional ASR applications for widely spoken languages with SPRAAK, the aim of the software is not to compete against well-known commercial packages with a large installed base. Rather we want to allow companies to build and implement ASR based solutions for niche markets. Example applications include but are not limited to the following:

- Speech transcription/alignment (“conventional ASR applications”) for low resourced or under-resourced languages, e.g. small languages in both developed countries (e.g. Frisian) and developing countries (e.g. Afrikaans);
- Speech segmentation, e.g. for speech corpus development;
- Subtitling;
- Medical applications e.g. speech evaluation for patients with speech disorders;
- Educational applications e.g. reading tutors, pronunciation training;
- Phonetic labeling in phonetics research;
- Speaker recognition.

For some of the listed tasks a set of scripts are provided. We expect the number of available scripts to grow rapidly over time as the user base of SPRAAK grows.

6.2.4 High Level API for the Run-Time Engine

The high level API is the convenient interface to communicate with the SPRAAK run-time engine. It allows for all necessary functionality: start and stop, setting of parameters, loading and switching of acoustic and language model and control over the audio device. This level of control suits the needs of people interested

in dialog development, interactive recogniser behaviour, speech recognition testing from large corpora, interactive testing, building demonstrators, etc.

The high level API is available in two forms: C-callable routines and a corresponding set of commands for a client-server architecture. There is a one-to-one mapping between these two forms.

In the client-server framework, client (application) and server (speech recognition engine) communicate with each other via a simple physical interface using the high level API commands. Feedback from the engine to the application is both client driven and server driven. All concepts known to the high level API may also be specified in an initialisation file that is loaded when the engine is started.

The alternative form of the high level API uses C-callable routines instead of the pipe/socket interface layer. Except for giving up the flexibility of the client/server concept, it has exactly the same functionality. This is the most compact and efficient implementation and would be the implementation of choice for (semi)-commercial applications on a standalone platform.

The high level API gives the user an intuitive and simple access to a speech recognition engine, while maintaining reasonably detailed control. The concepts defined in the high level API are concepts known to the speech recognition engine such as language models, acoustic models, lexica, pre-processing blocks and search engines. In that sense the SPRAAK high level API may not be as high level as in certain commercial packages which tend to work with concepts such as ‘user’, ‘language’, ‘application’. If a translation is necessary from intuitive user concepts to the speech recognition concepts then this is the task of the application, although SPRAAK provides the necessary hooks to work with these high level concepts.

Usage of the high level API should be suitable for users with a moderate understanding of speech recognition concepts and requires only a minimum of computational skills.

6.2.5 Low Level API

The low level API provides full access to all routines, including those that are not relevant for the run-time engine. In order to understand what is possible through this low level API it is necessary to have an understanding of the software architecture of the SPRAAK toolkit. In essence, the SPRAAK toolkit consists of a set of programs, modules (compiled code) and scripts. The modules are designed according to object oriented concepts and are written in C. Python is used as scripting language.

Standalone tools such as alignment, enrolment, operations on acoustic models, training of acoustic models are implemented in Python scripts that make use of the low level API. SPRAAK includes example scripts for most of these tasks, and the number of available scripts will grow over time.

Usage of the low level API gives the user full control over the internals. It is the vehicle for the ASR researcher who wants to write new training scripts, modify the recogniser, etc. New concepts may first be prototyped in Python. As

long as it doesn't affect the lowest level computational loops in the system the impact on efficiency will be limited. Nevertheless the inherent delays at startup time and the loss of efficiency in the scripting language make this setup less suited for applications.

The low level API is also the ideal tool for ASR teaching. It provides detailed enough insight into the speech recognition internals, it allows for visualisation of intermediate results and makes it possible to modify specific pieces of the code without the need to develop a full recognition system.

Usage of the low level API is intended for those who have a good understanding of speech recognition internals and who are at least skilful programmers at the script level.

6.2.6 SPRAAK Developers API

The SPRAAK developers API is mainly relevant to the developers of the SPRAAK toolkit. It handles low-level tasks such as parsing input, argument decoding, the object oriented layer, debug messages and asserts, ...

For a number of reasons, not all of the functionality of the developers API is made available on the Python level:

- Python provides its own alternatives (parsing, hash tables, the object oriented layer);
- The functionality is not relevant at the Python level (atomic operations);
- Exposing the functionality would be difficult and with little merit.

Routines that are part of the developers API only are tailor made to operate within the SPRAAK library and are (in theory) not fit for use outside this scope. Since these routines are only meant for internal use, their interface can also be changed without prior notice.

Extending and modifying the developers API is intended only for those who have a good understanding of the SPRAAK framework and who are skilful C-programmers.

6.3 Features of the SPRAAK Toolkit

6.3.1 Overview

Figure 6.1 shows the main components of the SPRAAK recogniser in a typical configuration and with the default interaction between these components.

In a large vocabulary speech recognition application typically all components will be activated. When using the SPRAAK engine for other tasks

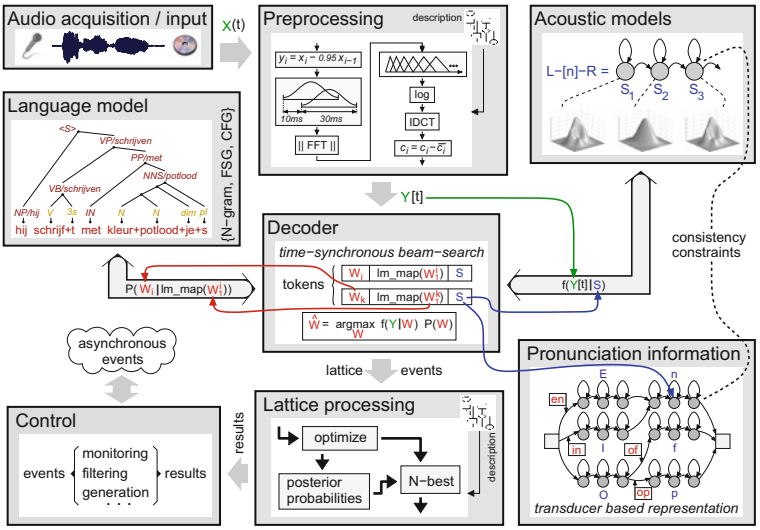


Fig. 6.1 Main components of the SPRAAK recogniser

(e.g. preprocessing, alignment, training of acoustic models, ...), only a subset of the same main components may be used. For highly experimental setups more complex configurations and interactions are possible.

The components shown in Fig. 6.1 are:

- The preprocessing;
- The acoustic model (AM);
- The lexicon containing pronunciation information, organised as a network/finite state transducer;
- The language model (LM);
- The decoder which connects all major components together;
- A lattice generation and processing block.

A detailed description of the underlying software implementation may be found in the developer's manual. Here we give a brief functional description of each of these modules.

6.3.2 The Preprocessing

Speech recognisers do not work directly on sampled data. SPRAAK as the great majority of speech recognition systems works with frames, i.e. features extracted from the data at regularly spaced time intervals.

SPRAAK has a wealth of signal processing routines and options built in. The algorithm to be executed is described by a scripting language in a signal processing

configuration file. It may be called ‘on-line’, i.e. processing is done as soon as input becomes available and the computed features are streamed towards further processing steps in an automatic way. This can be done both during training and recognition. This is the standard setup for most types of signal processing which require only minimal amounts of computing time on today’s systems. Processing may also be done ‘off-line’ in which the signal processing output is written to file. The latter may be the preferred mode of operation for complex signal processing algorithms.

A complete list of all signal processing modules can be found in the documentation. The implemented modules include standard speech processing blocks such as FFT, cepstra, mel-cepstra, LPC and PLP analysis, mean normalisation, histogram equalisation, vocal tract length normalisation, begin- and endpoint detection, time derivatives and silence/speech detection. The flexibility of the preprocessing is further extended with generic blocks such as linear transformations, linear filters, vector based function evaluation, frame stacking and order statistics. Support for neural network processing and gaussian mixture evaluation is also available.

Important properties and constraints of the preprocessing module are:

- Non-causal behaviour is supported by allowing the process routine to backhold data whenever deemed necessary;
- All preprocessing can be done both on-line and off-line;
- The fact that everything is available on-line is very handy, but requires some programming effort when writing new modules since everything has to be written to work on streaming data;
- Currently, only a single frame clock is supported. Changing the frame-rate (dynamically or statically) is not supported;
- Frames must be processed in order and are returned in order.

6.3.3 The Acoustic Model

The acoustic model calculates observation likelihoods for the Hidden Markov (HMM) states. These likelihoods can be provided by Gaussian mixture models, neural networks, discrete distributions, or can even be read from file. When using Gaussian mixture models, SPRAAK offers the unique possibility to share gaussians between states, on top of the traditional state tying. In our experience most tasks are served best with this unique modelling technique. In this setup each distribution in the acoustic model is computed as a weighted sum of gaussians drawn from a large pool. It is used as a generic approach that allows untied and fully tied mixtures as extremes of its implementation. Features are:

- Mixture gaussian densities with full sharing;
- Fast evaluation for tied Gaussians by data-driven pruning based on ‘Fast Removal of Gaussians’ (FRoG) [6];

- Model topology is decoupled from observation likelihoods, allowing for any number of states in any subword unit;
- Dedicated modules for initialising and updating the acoustic models (training and/or speaker adaptation);
- Access to all components of the acoustic model (the Gaussian set, FRoG, . . .);
- Implementations for discrete density and neural net models.

6.3.3.1 Tied Gaussian Mixtures

In continuous HMMs the model of the observation probabilities is organised in two layers, consisting of:

- A set of basis functions (typically multivariate gaussians) that are shared over all states;
- A set of weights (typically probabilities) linking the basis functions to the states.

The number of basis functions tends to be very large (several thousands), while only a few of them (typically around 100 in SPRAAK) will have non-zero weights for any individual state. Given this completely constraint-free tying scheme, other variants of tying can be accomodated: untied gaussians in which gaussians are private to a state, and fully tied gaussians in which for each state a weight is assigned to each gaussian. In this type of modelling the weight matrix linking basis functions to states tends to be very sparse (i.e. most weights are '0'). SPRAAK accommodates a flexible sparse data structure to cope with this efficiently.

The acoustic model returns normalised scores for a segment of data. By default the Viterbi algorithm is used, but other computational approaches and other acoustic models (than the standard HMM) are possible.

'Normalised' implies that the sum of scores over all possible states for any given time instant will be equal to 1.0. In practice log10-probabilities are returned. The motivation for using these normalised scores is:

- Normalised scores are much easier to interpret than unnormalised scores, whose values tend to fluctuate a lot under varying external conditions such as background noise, speaker identity, . . .
- These normalised scores are not posteriors, however, as the prior state probabilities are used and not the current state probabilities (which are expensive to compute);
- The normalisation is identical for all hypothesis, hence it does not affect the ranking of different hypotheses; nor does it affect the outcome of training;
- These normalised scores are a convenient input for the computation of confidence scores.

SPRAAK uses an efficient bottom up scheme to predict which gaussians in the pool need to be evaluated and which ones not ("Fast Removal Of Gaussians" – FRoG). This is done for the whole pool of gaussians at once and not on a state by

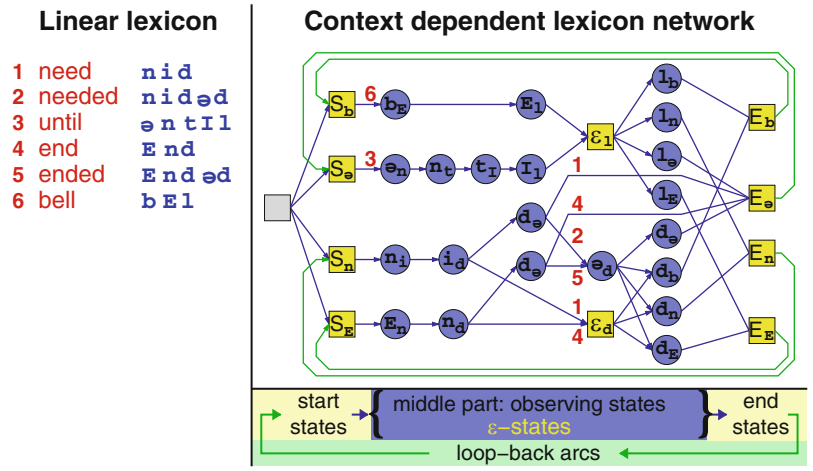


Fig. 6.2 Lexical network

state basis. The data structure describing for each axis which regions are relevant for which gaussians is computed once when the acoustic model is loaded.

6.3.4 The Lexicon and Pronunciation Network

The lexicon is stored as a pronunciation network in a (possibly cyclic) finite state transducer (FST) that goes from HMM states (input symbols of the FST) to some higher level (the output symbols of the FST). Typically, the output symbols are words. Having phones as output symbols is possible and results in a phone recogniser.

Apart from (word) pronunciations as such, this network can also encode assimilation rules and may use context dependent phones as learned by the acoustic model.

The same network may also be used to implement the constraints imposed by an FST-based language model. For example when training the acoustic models, the sentence being spoken (a linear sequence of words) is directly encoded in the pronunciation network, eliminating the need for a language model component.

Figure 6.2 gives an example of a pronunciation network using right context dependent phones and no assimilation rules. The final pronunciation network used by the decoder will also incorporate the tied-state information coming from the acoustic model. In order to not obscure the figure, this information was left out in the example.

The pronunciation network deviates from normal FST's in several ways:

- The input symbols are attached to the states, not the arcs;
- The network also contains some non-observing states: end-states, start-states and ϵ-states (non emitting states that allow more compact networks, cf. Fig. 6.2);

- The network contains two levels of output symbols: the phone identities and the items known by the language model.

6.3.5 The Language Model

The language model (LM) calculates conditional word probabilities, i.e. the probability of a new word given its predecessor words. For efficiency reasons the LM condenses all relevant information concerning the word predecessors in its own state variable(s).

Supported LM's and interfaces are:

- A word based N-gram which has low memory footprint and is fast. SPRAAK does not have its own tools for computing statistical language models but interfaces well with models generated with well established open source toolkits such as the SRI or CMU Language Model toolkits [2, 21, 22]. The language models produced by the mentioned toolkits come in the '.arpabo'-format (N-grams with backoff). In SPRAAK this format is converted to a compact binary format for efficiency reasons.
- A finite state grammar/transducer (FSG/FST). The FSG supports on the fly composition and stacking. The FSG has also provisions to behave exactly as an N-gram (correct fallback) and thus can replace the N-gram in situations where on-line changes to the LM are needed.

Furthermore an extension layer on top of these LM's allows for various extensions:

- Making class based LM's;
- Adding new words that behave similarly to existing words;
- Allowing multiple sentences to be uttered in one go;
- Adding filler words;
- Adding sub-models, e.g. a phone loop model to model the out-of-vocabulary words.

6.3.6 The Decoder

The decoder (search engine) finds the best path through the search space defined by the acoustic model, the language model and the pronunciation network given the acoustic data coming from the preprocessing block.

SPRAAK implements an efficient all-in-one decoder with as main features:

- Breadth-first frame synchronous;
- Allows cross-word context-dependent tied-state phones, multiple pronunciations per word, assimilation rules, and any language model that can be written in a left-to-right conditional form;

- Exact, i.e. no approximations whatsoever are used during the decoding, except for the applied pruning;
- Pruning: a threshold is used to discard hypotheses whose score falls a certain amount below the score of the most likely hypothesis; if most hypotheses have a similar score, a beam width parameter is used to keep only the best hypotheses;
- Provides as well the single best output as (word) lattices. Both outputs can be generated on-the-fly and with low latency;
- The backtracking can be instructed to keep track of the underlying phone or state sequences and to add them to the recognised word string and/or store them alongside the (word) lattice;
- The speed of the decoder is very high, e.g. depending on the complexity, a single pass in the model training is one or several orders of magnitude faster than realtime on a modern machine.

6.3.7 (Word) Lattice (Post)-processing

The (word) lattice (post)-processing consists, similar to the preprocessing, of a fully configurable processing flow-chart and a large set of processing blocks. The lattice processing can be fed either with stored lattices or can be coupled directly to the decoder using the built-in lattice generator.

The most important properties of the lattice processing component are:

- A low-latency data driven design suitable for use in real-time applications;
- Lattices that contain only acoustic model scores;
- Weak consistency checks when re-scoring for speed reasons (this may result in crashes if inconsistent knowledge sources are applied).

Among the available processing modules are:

- Lattice re-scoring (new LM) using a pseudo frame synchronous (breadth-first) decoder; this decoder is low-latency;
- The FLaVoR-decoder [8]. This decoder goes from phone lattices to word lattices using a fast and robust decoder. Its primary goal is to allow the use of advanced linguistic models (morpho-phonology, morpho-syntax, semantics, . . .);
- Calculating posterior probabilities given the (word) lattice and some finite state based language model. This is also the module that can be used to introduce language model constraints (and scores) in the lattice;
- Searching the best fit between two lattices, i.e. finding the best path through an input lattice given a reference lattice and a substitution matrix;
- A check module that verifies whether the input lattice adheres to requirements set forth by the different processing blocks;
- Input and output modules;
- Several optimisation modules;

- Lattices can also be converted to the HTK Standard Lattice Format (SLF), hence allowing the use of other external toolkits for operations such as N-best lists, confidence scoring via word posteriors, consensus networks, . . .

6.3.8 Extension to a Missing Data Theory Based Recogniser

In order to improve its robustness to noise and possibly reverberation, SPRAAK has been extended to include a Missing Data Theory (MDT) algorithm. This is the result of the STEVIN MIDAS project – cf. Chap. 16, p. 289. It does NOT improve robustness to pronunciation variation or accents.

The implementation is largely based on [27]. MDT requires an additional component, a mask estimator, which indicates which data is reliable enough for the recogniser to use and which data is unreliable (and will be reconstructed). Its output is a mask, a vector of variables that reveal to which extent each frequency channel is reliable (either binary or as a continuous variable between 0 and 1). Reliable data is merely copied from the input, while unreliable data is reconstructed or *imputed* based on the clean speech model. In the SPRAAK implementation, this imputation is guided by the search process, i.e. the missing data is reconstructed for each assumption the back-end makes about the speech identity. In this context, assumption means the visited state and selected Gaussian from its mixture. The missing data is reconstructed based on the maximum likelihood principle, i.e. it looks for the most likely values of the unreliable spectral components. This boils down to a constrained optimisation process that is solved with a gradient descent method, which is computationally efficient and requires only a few iterations to converge to a sufficiently accurate solution.

To use this MDT extension, the user needs to be aware of the following:

- It is assumed that the speech signal is disturbed by additive possibly non-stationary background noise. Significant reverberation cannot be handled, though mask estimation methods to handle reverberated speech are described in the literature [18, 19]. If the user wants robustness against reverberation, he will need to implement his own mask estimation technique.
- Tests on noisy data have shown that the MDT recogniser runs at about the same speed as when SPRAAK runs a standard MIDA front-end [7], but at a higher accuracy. Notice that noisy data requires longer decoding times due to the increased ambiguity.
- The MDT recogniser is implemented as an extension: the user needs to provide a mask estimator and additional data components, but the existing acoustic model, language model, lexicon, . . . remain unchanged.

6.4 SPRAAK Performance

During its history, SPRAAK and its predecessor have been used in many projects and for many tasks. It has also been benchmarked with internationally used and well-known reference tasks. Benchmark results achieved with SPRAAK include:

- TIDigits: 0.17 % WER using variable-sized word models (11–17 states).
- WSJ 5k closed vocabulary, speaker independent, WSJ-0 acoustic training data, bigram LM: 4.8 % WER on the nov92 test set (this is our AURORA-4 clean speech reference). When using more acoustic data (WSJ-1) and a trigram, a WER of 1.8 % can be achieved [24].
- WSJ 20k open vocabulary (1.9 % OOV rate), speaker independent, WSJ-1 acoustic training data, trigram: 7.3 % WER on the nov92 test set. This task runs in real time [11].
- Switchboard spontaneous telephone conversations, 310 h of acoustic training data, 3M words for training of the LM, open vocabulary: 29 % WER on the 2001 HUB5 benchmark [23]. This result was obtained with a single decoding pass and without any speaker adaptation, running at $4 \times$ RT on a pentium4 2.8 GHz. A similar real-time configuration gives a WER of 31 %.
- Dutch NBest Benchmark: 20.3 % WER on the Flemish Broadcast News test set [9].

Note that performances depend not only on the software but also very much on the models. A fair comparison of SPRAAK to other speech recognizers is therefore not easy if one wants to know the influence of the software only and not of the difference in the models. Processing times are also very volatile since hardware speed increases continuously and results on processing time reported some time ago cannot be compared to recent results. Therefore we opted not to try to include results from other systems.

On machines with multi-core processors, training of acoustic models can be executed in parallel, allowing a faster turn-around time when developing new models and applications. The goal of the SPRAAK developers is to extend the amount of code that can take profit from parallel hardware.

6.5 SPRAAK Requirements

SPRAAK is developed on the Linux platform and therefore all components are available on Unix(-like) operating systems, including Mac OS-X. On Windows XP or higher, a run-time engine and most standalone programs are available (though with some limitations); automatic parallelisation of some scripts and certain advanced functionalities (distributed computing and resource sharing, mainly relevant for resource development) are not available at this time.

Essential tools for installing and running SPRAAK include:

- A C99 compliant compiler for compilation (gcc recommended);
- Python (version 2.5 or higher) for installation and at run-time;
- Scons (version 0.98 or higher) for software installation.

For extended functionality and to build the documentation, additional applications, libraries and header files are useful or needed, as explained on the SPRAAK website http://www.spraak.org.

6.6 SPRAAK Licensing and Distribution

At the conception of the SPRAAK project it was found that the major overhaul of the old existing code was a great opportunity to at the same time open up the code to the community at large. It was decided to make the software an open source package, free for use for academic purposes. Therefore an academic license is available that is free of charge and available to research institutes or universities upon request. Before giving access to the software, a duly signed license agreement is required, which can be generated by registering on SPRAAK's website http://www.spraak.org/. A license key and instructions are sent out after reception of the agreement.

For companies, an evaluation license is available at low cost, that allows the evaluation of the software during a period of 1 year. A license for commercial use is also available. The aim of the software is not to compete against well-known commercial packages with a large installed base but rather to allow companies to build and implement ASR based solutions for niche markets. Revenues from commercial contracts are reserved for future upgrades of the SPRAAK software.

6.7 SPRAAK in the STEVIN Programme

As already mentioned, it was stated at the conception of the STEVIN programme that the availability of a speech recognition system for Dutch was one of the essential requirements for the language and speech technology community. Now that SPRAAK is available and that the STEVIN programme has ended we can verify how the software was put to use in other STEVIN projects. This is a list of projects that have some relation to SPRAAK:

- JASMIN-CGN: in this corpus project, SPRAAK and its predecessor HMM75 have generated phonetic annotations and word segmentations – cf. [4] and Chap. 3.
- MIDAS (Missing Data Solutions): this project tackles the noise robustness problem in ASR through missing data techniques (MDT). The results of the project have been integrated in the software – cf. Sect. 6.3.8, Chap. 16 and [13].
- NEON (Nederlandstalige Ondertiteling): this project evaluates the use of ASR in subtitling applications. SPRAAK was used as the ASR component – cf. [28].

- DIADEMO: this project has built a demonstrator that recognizes spoken dialects. SPRAAK is used as the ASR component. The algorithm is explained in [29].
- DISCO (Development and Integration of Speech technology into COurseware for language learning): this present project has developed and tested a prototype of an ASR-based CALL application for training oral proficiency for Dutch as a second language. SPRAAK was used as the ASR component – cf. [26] and Chap. 18.
- AAP (Alfabetisering Anderstaligen Plan): a demonstrator was built that uses speech technology to combat illiteracy for second language learners of Dutch – cf. [25].
- HATCI (Hulp bij Auditieve Training na Cochleaire Implantatie): this project uses automatic speech assessment to build an application to support rehabilitation therapy after cochlear implantation. SPRAAK was used for the speech assessment component – cf. [17].
- N-Best: this project has organised and executed an evaluation of large vocabulary speech recognition systems trained for Dutch (both Northern and Southern Dutch) in two evaluation conditions (Broadcast News and Conversational Telephony Speech). Two submissions to the evaluation have used SPRAAK to generate their results – cf. [9, 15] and Chap. 15.

6.8 Future Work

A package like SPRAAK is of course never complete. The aim is however to follow the main trends in current ASR research and to implement interesting new algorithms as they become available. The software also still lacks some methods that have been in existence since some time, such as more extensive speaker adaptation than the currently available Vocal Tract Length Normalisation (VTLN). The aim is to include to that end CMLLR (Constrained/Feature Space MLLR [1, 12]) in the short term. Also the following will be added:

- A high-level flow-chart scripting language to configure multi-stage recognisers, parallel recognisers with the available processing blocks;
- Further parallelisation of the code;
- Posterior probabilities on networks: this allows to build consensus networks where a consensus output is derived by selecting the word sequence with the best score, where scores can be formed in many ways such as by voting or using posterior probability estimates. Some external toolkits exist to produce these, starting from our lattices but they may fail for large lattices or with complex language models.

User contributions are also very welcome. The academic license terms specify that users who implement “modifications” (that change, improve or optimize an existing functionality) must make these available to the SPRAAK developers, and users who implement “new modules” (adding new functionality) must inform the developers

about them and make them available at reasonable conditions under the form of a non-exclusive license. This mechanism will hopefully contribute to the capabilities of the software and strengthen of the user base.

6.9 Conclusions

The STEVIN SPRAAK project has addressed one of the essential requirements for the language and speech technology community: the availability of a speech recognition system for Dutch. Researchers can use the highly modular toolkit for research into speech recognition algorithms (language independent and thus not only for Dutch). It allows them to focus on one particular aspect of speech recognition technology without needing to worry about the details of the other components. Besides that a state-of-the art recogniser for Dutch with a simple interface is now available, so that it can be used by non-specialists with a minimum of programming requirements as well. Next to speech recognition, the resulting software also enables applications in related fields. Examples are linguistic and phonetic research where the software can be used to segment large speech databases or to provide high quality automatic transcriptions. The existing ESAT recogniser was augmented with knowledge and code from the other partners in the project, as a starting point, and this code base was transformed to meet the specified requirements. The transformation was accomplished by improving the software interfaces to make the software package more user friendly and adapted for usage in a large user community, and by providing adequate user and developer documentation written in English, so as to make it easily accessible to the international language and speech technology community as well. The software is open source and freely available for research purposes. Details, documentation and references can be found at http://www.spraak.org/. Example recognizers, demo acoustic models for Dutch and their training scripts can be obtained from the Dutch-Flemish HLT Central.

Acknowledgements The SPRAAK toolkit has emerged as the transformation of KULeuven/E-SAT's previous speech recognition system (HMM75 was the latest version). This system is the result of 20 years of research and development in speech recognition at the KULeuven. We want to acknowledge the contributions of many researchers (too many to list them all) most of which have by now left the university after having obtained their PhD degree.

The SPRAAK project has also added some extra functionality to the previous system and has produced demo acoustic models and documentation. This was done in a collaborative effort with four partners: KULeuven/ESAT, RUNijmegen/CSLT, UTwente/HMI and TNO. The contributions of all these partners are also acknowledged.

References

1. Afify, M., Siohan, O.: Constrained maximum likelihood linear regression for speaker adaptation. In: Proceedings of the ICSLP-2000, Beijing, China, vol.3, pp. 861–864 (2000)
2. Clarkson, P., Rosenfeld, R.: Statistical language modeling using the CMU-Cambridge toolkit. In: Proceedings of the ESCA Eurospeech 1997, Rhodes, Greece, pp. 2707–2710 (1997)
3. CMUSphinx wiki available at. http://cmusphinx.sourceforge.net/wiki/start
4. Cucchiarini, C., Driesen, J., Van hamme, H., Sanders, E.: Recording speech of children, non-natives and elderly people for HLT applications: the JASMIN-CGN corpus. In: Proceedings of the 6th International Conference on Language Resources and Evaluation – LREC 2008, Marrakech, Morocco, 28–30 May 2008, p. 8
5. Demuynck, K.: Extracting, modelling and combining information in speech recognition. Ph.D. thesis, K.U.Leuven ESAT (2001)
6. Demuynck, K., Duchateau, J., Van Compernolle, D.: Reduced semi-continuous models for large vocabulary continuous speech recognition in Dutch. In: Proceedings of the ICSLP, Philadelphia, USA, Oct 1996, vol. IV, pp. 2289–2292
7. Demuynck, K., Duchateau, J., Van Compernolle, D.: Optimal feature sub-space selection based on discriminant analysis. In: Proceedings of the European Conference on Speech Communication and Technology, Budapest, Hungary, vol. 3, pp. 1311–1314 (1999)
8. Demuynck, K., Laureys, T., Van Compernolle, D., Van hamme, H.: FLaVoR: a flexible architecture for LVCSR. In: Proceedings of the European Conference on Speech Communication and Technology, Geneva, Switzerland, Sept 2003, pp. 1973–1976
9. Demuynck, K., Puurula, A., Van Compernolle, D., Wambacq, P.: The ESAT 2008 system for N-Best Dutch speech recognition Benchmark. In: Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, Merano, Italy, Dec 2009, pp. 339–343
10. Demuynck, K., Roelens, J., Van Compernolle, D., Wambacq, P.: SPRAAK: an open source speech recognition and automatic annotation kit. In: Proceedings of the International Conference on Spoken Language Processing, Brisbane, Australia, Sept 2008, p. 495
11. Demuynck, K., Seppi, D., Van Compernolle, D., Nguyen, P., Zweig, G.: Integrating meta-information into exemplar-based speech recognition with segmental conditional random fields. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, May 2011, pp. 5048–5051
12. Gales, M.: Maximum likelihood linear transformations for HMM-based speech recognition. Comput. Speech Lang. **12**, 75–98 (1998)
13. Gemmeke, J.F., Van Segbroeck, M., Wang, Y., Cranen, B., Van hamme, H.: Automatic speech recognition using missing data techniques: handling of real-world data. In Kolossa, D., Haeb-Umbach, R. (eds.) Robust Speech Recognition of Uncertain or Missing Data. Springer, Berlin/Heidelberg (2011, in press)
14. HTK website. http://htk.eng.cam.ac.uk/
15. Kessens, J., van Leeuwen, D.A.: N-best: the Northern- and Southern-Dutch benchmark evaluation of speech recognition technology. In: Proceedings of the Interspeech, Antwerp, Belgium, pp. 1354–1357 (2007)
16. Lee, A., Kawahara, T.: Recent development of open-source speech recognition engine Julius. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Sapporo, Japan, pp. 131–137 (2009)
17. Nogueira, W., Vanpoucke, F., Dykmans, P., De Raeve, L., Van hamme, H., Roelens, J.: Speech recognition technology in CI rehabilitation. Cochlear Implant. Int. **11**(Suppl. 1), 449–453 (2010)
18. Palomäki, K., Brown, G., Barker, J.: Missing data speech recognition in reverberant conditions. In: Proceedings of the ICASSP, Orlando, Florida, pp. 65–68 (2002)
19. Palomäki, K., Brown, G., Barker, J.: Techniques for handling convolutional distortion with 'missing data' automatic speech recognition. Speech Commun. **43**(1–2), 123–142 (2004)

20. Rybach, D., Gollan, C., Heigold, G., Hoffmeister, B., Lööf, J., Schlüter, R., Ney, H.: The RWTH Aachen university open source speech recognition system. In: Proceedings of the Interspeech 2009, Brighton, UK, Sept 2009, pp. 2111–2114
21. Stolcke, A.: SRILMan extensible language modeling toolkit. In: Hansen, J.H.L., Pellom, B. (eds.) Proceedings of the ICSLP, Denver, Sept 2002, vol. 2, pp. 901–904
22. Stolcke, A., Zheng, J., Wang, W., Abrash, V.: SRILM at sixteen: update and outlook. In: Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, Waikoloa, Hawaii, Dec 2011
23. Stouten, F., Duchateau, J., Martens, J.-P., Wambacq, P.: Coping with disfluencies in spontaneous speech recognition: acoustic detection and linguistic context manipulation. Speech Commun. **48**(11), 1590–1606 (2006)
24. Stouten, V., Van hamme, H., Wambacq, P.: Model-based feature enhancement with uncertainty decoding for noise robust ASR. Speech Commun. **48**(11), 1502–1514 (2006)
25. Strik, H., Bakker, A.: Alfabetisering met een luisterende computer. DIXIT special issue on 'STEVIN en onderwijs' (in Dutch), p. 21. http://lands.let.ru.nl/~strik/publications/a148-TST-AAP-DIXIT.pdf (2009)
26. van Doremalen, J., Cucchiarini, C., Strik, H.: Optimizing automatic speech recognition for low-proficient non-native speakers. EURASIP J. Audio Speech Music Process. **2010**, 13 (2010). Article ID 973954. doi:10.1155/2010/973954
27. Van Segbroeck, M., Van hamme, H.: Advances in missing feature techniques for robust large vocabulary continuous speech recognition. IEEE Trans. Audio Speech Lang. Process. **19**(1), 123–137 (2011)
28. Wambacq, P., Demuynck, K.: Efficiency of speech alignment for semi-automated subtitling in Dutch. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNAI, vol. 6836, pp. 123–130. Springer, Berlin/New York (2011)
29. Wu, T., Duchateau, J., Martens, J.-P., Van Compernolle, D.: Feature subset selection for improved native accent identification. Speech Commun. **52**, 83–98 (2010)

Chapter 7
COREA: Coreference Resolution for Extracting Answers for Dutch

Iris Hendrickx, Gosse Bouma, Walter Daelemans, and Véronique Hoste

7.1 Introduction

Coreference resolution is essential for the automatic interpretation of text. It has been studied mainly from a linguistic perspective, with an emphasis on the recognition of potential antecedents for pronouns. Many practical NLP applications such as information extraction (IE) and question answering (QA), require accurate identification of coreference relations between noun phrases in general. In this chapter we report on the development and evaluation of an automatic system for the robust resolution of referential relations in text. Computational systems for assigning such relations automatically, require the availability of a sufficient amount of annotated data for training and testing. Therefore, we annotated a Dutch corpus of 100K words with coreferential relations, and in addition we developed guidelines for the manual annotation of coreference relations in Dutch.

I. Hendrickx (✉)
Centro de Linguística da Universidade de Lisboa, Faculdade das Letras, Av. Prof. Gama Pinto 2, 1649-003, Lisboa, Portugal
e-mail: iris@clul.ul.pt

G. Bouma
Information Science, University of Groningen, Oude Kijk in 't Jatstraat 26, Postbus 716, NL 9700 AS, Groningen, The Netherlands
e-mail: g.bouma@rug.nl

W. Daelemans
CLiPS, University of Antwerp, Prinsstraat 13, 2000 Antwerpen, Belgium
e-mail: walter.daelemans@ua.ac.be

V. Hoste
LT3, School of Translation Studies, University College Ghent and Faculty of Linguistics, Ghent University, Groot-Brittanniëlaan 45, B-9000, Gent, Belgium
e-mail: veronique.hoste@hogent.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_7,

We evaluated the automatic coreference resolution module in two ways. On the one hand we used the standard internal approach to evaluate a coreference resolution system by comparing the predictions of the system to a hand-annotated gold standard test set. On the other hand we performed an application-oriented evaluation of our system by testing the usefulness of coreference relation information in an NLP application. We ran experiments with a relation extraction module for the medical domain, and measured the performance of this module with and without the coreference relation information. In a separate experiment we also evaluated the effect of coreference information produced by another simple rule-based coreference module in a question answering application.

The chapter is structured as follows. We first summarise related work in Sect. 7.2. We present the corpus that is manually annotated with coreference relations in Sect. 7.3.1. Section 7.3.2 details the automatic coreference resolution system and in Sect. 7.4 we show the results of both the internal and the application-oriented evaluation. We conclude in Sect. 7.5.

7.2 Related Work

In the last decade considerable efforts have been put in annotating corpora with coreferential relations. For English, many different data sets with annotated coreferential relations are available such as the MUC-6 [8] and MUC-7 [23] data sets, ACE-2 [7], GNOME corpus [26], ARRAU [27], and more recently, OntoNotes 3.0 [39]. But also for other languages data sets exist such as for German, the TBa-D/Z coreference corpus [12] and the Potsdam corpus [19], for Czech the Prague Dependency Treebank (PDT 2.0) [20], for Catalan AnCora-CO [29], for Italian I-CAB [22] and the Live Memories Corpus [31], and the Copenhagen Dependency Treebank [18] for Danish, English, German, Italian, and Spanish. Most of these corpora follow their own annotation scheme. In SemEval-2010, Task 1 *Coreference Resolution in Multiple Languages* was devoted to multi-lingual coreference resolution for the languages Catalan, Dutch, English, German, Italian and Spanish [30]. The CoNLL 2011 and 2012 shared tasks are also dedicated to automatic coreference resolution.

For Dutch, besides the COREA corpus described in Sect. 7.3.1 there is currently also a data set of written new media texts such as blogs [9] developed in the DuOMan project described in Chap. 20, page 359 and a substantial part (one million words) of the SoNaR corpus [32] (Chap. 13, page 219) is also annotated with coreference. All these data sets have been annotated according to the COREA annotation guidelines. For the Dutch language we can now count on a large, and rich data set that is suitable both for more theoretical linguistic studies of referring expressions and for practical development and evaluation of coreference resolution systems. By covering a variety of text genres, the assembled data set can even be considered as a unique resource for cross-genre research.

Currently there are not many coreference resolution systems for Dutch available. The first full-fledged system was presented by Hoste [14, 15] in 2005 and this is

the predecessor of the system described in Sect. 7.3.2. More recently, two of the participating systems in the SemEval-2010 Task 1 on multi-lingual coreference resolution were evaluated for all six languages including Dutch. The UBIU system [40], was a robust language independent system that used a memory-based learning approach using syntactic and string matching features. The SUCRE system [17] obtained the overall best results for this SemEval task and used a more flexible and rich feature construction method and a relational database in combination with machine learning.

7.3 Material and Methods

7.3.1 Corpus and Annotation

The COREA corpus is composed of texts from the following sources:

- Dutch newspaper articles gathered in the DCOI project[1]
- Transcribed spoken language material from the Spoken Dutch Corpus (CGN)[2]
- Lemmas from the Spectrum (Winkler Prins) medical encyclopedia as gathered in the IMIX ROLAQUAD project[3]
- Articles from KNACK [16], a Flemish weekly news magazine.

All material from the first three sources was annotated in the COREA project. The material from KNACK was already annotated with coreference relations in a previous project (cf. [14]). Note that the corpus covers a number of different genres (speech transcripts, news, medical text) and contains both Dutch and Flemish sources. The latter is particularly relevant as the use of pronouns differs between Dutch and Flemish [36].

The size of the various subcorpora, and the number of annotated coreference relations is given in Table 7.1.

For the annotation of coreference relations we developed a set of annotation guidelines [3] largely based on the MUC-6 [8] and MUC-7 [23] annotation scheme for English. Annotation focuses primarily on coreference or IDENTITY relations between noun phrases, where both noun phrases refer to the same extra-linguistic entity. These multiple references to the same entity can be regarded as a *coreferential chain* of references. While these form the majority of coreference relations in our corpus, there are also a number of special cases. A BOUND relation exists between an anaphor and a quantified antecedent, as in *Everybody$_i$ did what they$_i$ could*. A BRIDGE relation is used to annotate part-whole or set-subset relations, as in *the tournament$_i$ ... the quarter finals$_i$*. We also marked predicative (PRED) relations,

[1] DCOI: http://lands.let.ru.nl/projects/d-coi/

[2] CGN: http://lands.let.ru.nl/cgn/

[3] IMIX: http://ilk.uvt.nl/rolaquad/

Table 7.1 Corpus statistics for the coreference corpora developed and used in the COREA project. IDENT, BRIDGE, PRED and BOUND refer to the number of annotated identity, bridging, predicative, and bound variable type coreference relations respectively

Corpus	DCOI	CGN	MedEnc	Knack
#docs	105	264	497	267
#tokens	35,166	33,048	135,828	122,960
# IDENT	2,888	3,334	4,910	9,179
# BRIDGE	310	649	1,772	na
# PRED	180	199	289	na
# BOUND	34	15	19	43

as in *Michiel Beute$_i$ is a writer$_i$*. Strictly speaking, these are not coreference relations, but we annotated them for a practical reason. Such relations express extra information about the referent that can be useful for example for a question answering application. We used several attributes to indicate situations where a coreference relation is in the scope of negation, is modified or time dependent, or refers to a meta-linguistic aspect of the antecedent.

Annotation was done using the MMAX2 tool.[4] For the DCOI and CGN material, manually corrected syntactic dependency structures were available. Following the approach of [12], we used these to simplify the annotation task by creating an initial set of markables beforehand. Labeling was done by several linguists.

To estimate the inter-annotator agreement for this task, 29 documents from CGN and DCOI were annotated independently by two annotators, who marked 517 and 470 coreference relations, respectively. For the IDENT relation, we compute inter-annotator agreement as the F-measure of the MUC-scores [38] obtained by taking one annotation as 'gold standard' and the other as 'system output'. For the other relations, we compute inter-annotator agreement as the average of the percentage of *anaphor-antecedent* relations in the gold standard for which an *anaphor-antecedent′* pair exists in the system output, and where *antecedent* and *antecedent′* belong to the same cluster (w.r.t. the IDENT relation) in the gold standard. Inter-annotator agreement for IDENT is 76 % F-score, for bridging is 33 % and for PRED is 56 %. There was no agreement on the three BOUND relations marked by each annotator. The agreement score for IDENT is comparable, though slightly lower, than those reported for comparable tasks for English and German [13, 37]. Poesio and Vieira [28] reports 59 % agreement on annotating 'associative coreferent' definite noun phrases, a relation comparable to our BRIDGE relation.

The main sources of disagreement were cases where one of the annotators fails to annotate a relation, where there is confusion between PRED or BRIDGE and IDENT, and various omissions in the guidelines (i.e. whether to consider headlines and other leading material in newspaper articles as part of the text to be annotated).

[4] http://mmax2.sourceforge.net/

7.3.2 Automatic Resolution System

We developed an automatic coreference resolution tool for Dutch [14] that follows the pairwise classification method of potential anaphora-antecedent pairs similar to the approach of Soon et al. [33]. As supervised machine learning method we decided to use memory-based learning. We used the Timbl software package (version 5.1) [4] that implements several memory-based learning algorithms.

As we used a supervised machine learning approach to coreference resolution the first step was to train the classifier on examples of the task at hand: texts with manually annotated coreference relations. These manually annotated texts needed to be transformed into training instances for the machine learning classifier. First the raw texts were preprocessed to determine the noun phrases in the text and to gather grammatical, positional, and semantic information about these nouns. This preprocessing step involved a cascade of NLP steps such as tokenisation, part-of-speech tagging, text chunking, named entity recognition and grammatical relation finding as detailed in Sect. 7.3.3.

On the basis of the preprocessed texts, training instances were created. We considered each noun phrase (and pronoun) in the text as a potential anaphor for which we needed to find its antecedent. We processed each text backward, starting with the last noun phrase and pairing it with each preceding noun phrase, with a restriction of 20 sentences backwards. Each pair of two noun phrases was regarded as a training instance for the classifier. If a pair of two noun phrases belonged to the same manually annotated coreferential chain, it got a positive label; all other pairs got a negative label. For each pair a feature vector was created to describe the noun phrases and their relation (detailed in Sect. 7.3.4). Test instances were generated in the same manner. In total, 242 documents from the KNACK material were used as training material for the coreference resolution system.

The output from the machine learning classifier was a set of positively classified instances. Instead of selecting one single antecedent per anaphor (such as for example [25, 33]), we tried to build complete coreference chains for the texts and reconstruct these on the basis of the positive instances. As we paired each noun phrase with every previous noun phrase, multiple pairs can be classified as positive. For example, we have a text about Queen Beatrix and her name is mentioned five times in the text. In the last sentence there is the pronoun "she" referring to Beatrix. So we have a coreferential chain in the text of six elements that all refer to the same entity Beatrix. If we create pairs with this pronoun and all previous noun phrases in the text, we will have five positive instances each encoding the same information: "she" refers to Beatrix. For the last mention of the name Beatrix, there are four previous mentions that also refer to Beatrix, leading to four positive instances. In total there are $5 + 4 + 3 + 2 + 1 = 15$ positive instances for this chain while we need a minimum of five pairs to reconstruct the coreferential chain. Therefore we needed a second step to construct the coreferential chains by grouping and merging the positively classified instances that cover the same noun phrases. We grouped pairs

together and computed their union. When the overlap was larger than 0.1 we merged the chains together (we refer to [10] for more details on the merging procedure).

7.3.3 Preprocessing

The following preprocessing steps were performed on the raw texts: First, tokenisation was performed by a rule-based system using regular expressions. Dutch named entity recognition was performed by looking up the entities in lists of location names, person names, organisation names and other miscellaneous named entities. We applied a part-of-speech tagger and text chunker for Dutch that used the memory-based tagger MBT [5], trained on the Spoken Dutch Corpus.[5] Finally, grammatical relation finding was performed, using a shallow parser to determine the grammatical relation between noun chunks and verbal chunks, e.g. subject, object, etc. The relation finder [34] was trained on the previously mentioned Spoken Dutch Corpus. It offered a fine-grained set of grammatical relations, such as modifiers, verbal complements, heads, direct objects, subjects, predicative complements, indirect objects, reflexive objects, etc. We used the predicted chunk tags to determine the noun phrases in each text, and the information created in the preprocessing phase was coded as feature vectors for the classification step.

7.3.4 Features

For each pair of noun phrases we constructed a feature vector representing their properties and their relation [14]. For each potential anaphor and antecedent we listed their individual lexical and syntactic properties. In particular, for each potential anaphor/antecedent, we encode the following information, mostly in binary features:

- Yes/no pronoun, yes/no reflexive pronoun, type of pronoun (first/second/third person or neutral),
- Yes/no demonstrative,
- Type of noun phrase (definite or indefinite),
- Yes/no proper name,
- Yes/no part of a named entity,
- Yes/no subject, object, etc., of the sentence as predicted by the shallow parser.

For the anaphor we also encoded its local context in the sentence as a window in words and PoS-tags of three words left and right of the anaphor. We represented the relation between the two noun phrases with the following features:

[5] http://lands.let.ru.nl/cgn

- The distance between the antecedent and anaphor, measured in noun phrases and sentences;
- Agreement in number and in gender between both;
- Are both of them proper names, or is one a pronoun and the other a proper name;
- Is there a complete string overlap, a partial overlap, a overlap of the head words or is one an abbreviation of the other.

One particularly interesting feature that we have explored was the usage of semantic clusters [35]. These clusters were extracted with unsupervised k-means clustering on the Twente Nieuws Corpus.[6] The corpus was first preprocessed by the Alpino parser [1] to extract syntactic relations. The top-10,000 lemmatised nouns (including names) were clustered into a 1,000 groups based on the similarity of their syntactic relations. Here are four examples of the generated clusters:

- {barrière belemmering drempel hindernis hobbel horde knelpunt obstakel struikelblok} (Eng: obstacle impediment threshold hindrance encumbrance hurddle knot obstacle)
- {Disney MGM Paramount PolyGram Time_Warner Turner Viacom }
- {Biertje borrel cocktail cola drankje glaasje kopje pilsje} (Eng:beer shot cocktail glass cup cola drink pils)
- {Contour schaduw schim schrikbeeld silhouet verhaallijn} (Eng:contour shade shadow chimera silhouette story-line)

For each pair of referents we constructed three features as follows. For each referent the lemma of the head word was looked up in the list of clusters. The number of the matching cluster, or 0 in case of no match, was used as the feature value. We also constructed two features presenting the cluster number of each referent and a binary feature marking whether the head words of the referents occur in the same cluster or not.

In the first version of the coreference resolution system we coded syntactic information as predicted by the memory-based shallow parser in our feature set of 47 features [14]. In the COREA project we also investigated whether the richer syntactic information of a full parser would be a helpful information source for our task [11]. We used the Alpino parser [1], a broad-coverage dependency parser for Dutch to generate the 11 additional features encoding the following information:

- Named Entity label as produced by the Alpino parser, one for the anaphor and one for the antecedent.
- Number agreement between the anaphor and antecedent, presented as a four valued feature (values: *sg, pl, both, measurable_nouns*).
- Dependency labels as predicted for (the head word of) the anaphor and for the antecedent and whether they share the same dependency label.
- Dependency path between the governing verb and the anaphor, and between the verb and antecedent.

[6] http://www.vf.utwente.nl/$\sim$druid/TwNC/TwNC-main.html

- Clause information stating whether the anaphor or antecedent is part of the main clause or not.
- Root overlap encodes the overlap between 'roots' or lemmas of the anaphor and antecedent. In the Alpino parser, the root of a noun phrase is the form without inflections. Special cases were compounds and names. Compounds are split[7] and we used the last element in the comparison. For names we took the complete strings.

In total, each feature vector consisted of 59 features. In the next section we describe how we selected an optimal feature set for the classification and the results of the automatic coreference resolution experiments with and without these deeper syntactic features.

7.4 Evaluation

We performed both a direct evaluation and an external, application-oriented evaluation. In the direct evaluation we measured the performance of the coreference resolution system on a gold-standard test set annotated manually with coreference information. In the application-oriented evaluation we tried to estimate the usefulness of the automatically predicted coreference relations for NLP applications.

7.4.1 Direct Evaluation

Genetic algorithms (GA) have been proposed [6] as an useful method to find an optimal setting in the enormous search space of possible parameter and feature set combinations. We ran experiments with a generational genetic algorithm for feature set and algorithm parameter selection of Timbl with 30 generations and a population size of 10.

In this experiment we used ten fold cross validation on 242 texts from Knack. The GA was run on the first fold of the ten folds as running the GA is rather time-consuming. The found optimal setting was then used for the other folds as well. We computed a baseline score for the evaluation of the complete coreference chains. The baseline assigned each noun phrase in the test set its most nearby noun phrase as antecedent.

The results are shown in Table 7.2. Timbl scores well above the baseline in terms of F-score but the baseline has a much higher recall. The differences in F-score at the instance level between the model without and with syntactic features, are small,

[7]The Alpino parser uses various heuristics to determine whether words that are not in its dictionary can be analyzed as compounds. The most important heuristic splits a word in two parts where both parts must be in the dictionary, and the split that gives the longest suffix is chosen.

Table 7.2 Micro-averaged F-scores at the instance level and MUC F-scores at the chain level computed in ten fold cross validation experiments. Timbl is run with the settings as selected by the genetic algorithm (GA) without and with the additional Alpino features

Scoring at the instance level			
	Recall	Precision	F-score
TIMBL, GA	44.8	70.5	54.8
TIMBL, GA, with syntax	48.4	64.1	55.1
MUC scoring at the chain level			
	Recall	Precision	F-score
Baseline	81.1	24.0	37.0
TIMBL, GA	36.8	70.2	48.2
TIMBL, GA, with syntax	44.0	61.4	51.3

but when we look at the score computed at the chain level, we see an improvement of 3 % in F-score. Adding the additional features from the Alpino parser improves overall F-score by increasing the recall at the cost of precision.

7.4.2 Application-Oriented Evaluation

Below, we present the results of two studies that illustrate that automatic coreference resolution can have a positive effect on the performance of systems for information extraction and question answering.

7.4.2.1 Coreference Resolution for Information Extraction

To validate the effect of the coreference resolution system in a practical information extraction application, our industrial partner in this project, *Language and Computing NV*, constructed an information extraction module named *Relation Finder* which can predict medical semantic relations. This application was based on a version of the Spectrum medical encyclopedia (MedEnc) developed in the IMIX ROLAQUAD project, in which sentences and noun phrases were annotated with domain specific semantic tags [21]. These semantic tags denote medical concepts or, at the sentence level, express relations between concepts. Example 7.1 shows two sentences from MedEnc annotated with semantic XML tags. Examples of the concept tags are *con_disease, con_person_feature* or *con_treatment*. Examples of the relation tags assigned to sentences are *rel_is_symptom_of* and *rel_treats*.

Example 7.1.

```
<rel_is_symptom_of id="20">
  Bij <con_disease id="2">asfyxie</con_disease> ontstaat een
```

```
    toestand van
    <con_sympton id="7">bewustzijnverlies</con_sympton>
    en <con_disease id="4>shock</con_disease> (nauwelijks
    waarneembare
    <con_person_feature id="8">polsslag</con_person_feature> en
    <con_body_function id="13">ademhaling</con_body_function>).
  </rel_is_symptom_of>
  <rel_treats id="19">
    Veel gevallen van <con_disease id="6">asfyxie</con_disease>
    kunnen door
    <con_treatment id="14">beademing</con_treatment>, of
    door opheffen van de passagestoornis
    (<con_treatment id="15">tracheotomie</con_treatment>)
    weer herstellen.
  </rel_treats>
```

The core of the Relation Finder was a maximum entropy modeling algorithm trained on approximately 2,000 annotated entries of MedEnc. Each entry was a description of a particular item such as a disease or body part in the encyclopedia and contained on average ten sentences. It was tested on two separate test sets of 50 and 500 entries respectively. Our coreference module predicted coreference relations for the noun phrases in the data. We ran two experiments with the Relation Finder. In the first experiment we used the predicted coreference relations as features and the second one we did not use these features. On the small data set we obtained an F-score of 53.03 % without coreference and 53.51 % with coreference information. On the test set with 500 entries we got a slightly better score of 59.15 % F-score without and 59.60 % with coreference information. So for both test sets we observe a modest positive effect for the experiments using the coreference information.

7.4.2.2 Coreference Resolution for Question Answering

The question answering system for Dutch described in [2] used information extraction to extract answers to frequent questions off-line (i.e. the system tried to find all instances of the `capital` relation in the complete text collection off-line, to answer questions of the form *What is the capital of LOCATION?*). Tables with relation tuples were computed automatically for relations such as age of a person, location and date of birth, founder of an organisation, function of a person, number of inhabitants, winner of a prize, etc.

Using manually developed patterns, the precision of extracted relation instances is generally quite high, but coverage tends to be limited. One reason for this is the fact that relation instances are only extracted between entities (i.e. names, dates, and numbers). Sentences of the form *The village has 10,000 inhabitants* do not contain a ⟨*location,number_of_inhabitants*⟩ pair. If we can resolve the antecedent of *the village*, however, we can extract a relation instance.

To evaluate the effect of coreference resolution for this task, [24] extended the information extraction component of the QA system with a simple rule-based

Table 7.3 Number of relation instances, precision, and number of unique instances (facts) extracted using the baseline system, and using coreference resolution

	Instances	Precision	Facts
No coreference resolution	93,497	86 %	64,627
Pronoun resolution	3,915	40 %	3,627
Resolution on definite NPs	47,794	33 %	35,687
Total	145,141	72 %	103,941

coreference resolution system for pronouns. To resolve definite noun phrases, it used an automatically constructed knowledge base containing 1.3M class labels for named entities to resolve definite NPs.

Table 7.3 shows that, after adding coreference resolution, the total number of extracted facts went up with over 50 % (from 93K to 145K). However, the accuracy of the newly added facts was only 40 % for cases involving pronoun resolution and 33 % for cases involving definite NPs.

In spite of the limited accuracy of the newly extracted facts, we noticed that incorporation of the additional facts led to an increase in performance on the questions from the QA@CLEF 2005 test set of 5 % (from 65 to 70 %). We expect that even further improvements are possible by integrating the coreference resolution system described in Sect. 7.3.2.

7.5 Conclusion

Coreference resolution is useful in text mining tasks such as information extraction and question answering. Using coreference resolution, more useful information can be extracted from text, and that has a positive effect on the recall of such systems. However, it is not easy to show the same convincingly in application-oriented evaluations. The reason for this is that the current state-of-the-art in coreference resolution, based on supervised machine learning, is still weak, especially in languages like Dutch for which not a lot of training data is available. More corpora are needed, annotated with coreference relations.

We presented the main outcomes of the STEVIN COREA project, which was aimed at addressing this corpus annotation bottleneck. In this project, we annotated a balanced corpus with coreferential relations, trained a system on it, and carried out both a direct and application-oriented evaluations.

We discussed the corpus, the annotation and the inter-annotator agreement, and described the construction and evaluation of a coreference resolution module trained on this corpus in terms of the preprocessing and the features used.

We evaluated this coreference resolution module in two ways: with standard cross-validation experiments to compare the predictions of the system to a hand-annotated gold standard test set, and a more practically oriented evaluation to

test the usefulness of coreference relation information in information extraction and question answering. In both cases we observed a small but real positive effect of integrating coreference information, despite the relatively low accuracy of current systems. More accurate coreference resolution systems, should increase the magnitude of the positive effect. These systems will need additional semantic and world knowledge features. We showed the positive effect of richer syntactic features as generated by the Alpino parser, and of semantic features by means of the semantic cluster features we tested.

The annotated data, the annotation guidelines, and a web demo version of the coreference resolution system are available to all and are distributed by the Dutch TST HLT Agency.[8]

Acknowledgements We would like to thank Tim Van de Cruys for sharing his data sets of semantic clusters. We thank Anne-Marie Mineur, Geert Kloosterman, and Language & Computing for their collaboration in the COREA project.

References

1. Bouma, G., van Noord, G., Malouf, R.: Alpino: wide-coverage computational analysis of dutch. In: Computational Linguistics in the Netherlands 2000. Selected Papers from the Eleventh CLIN Meeting, Tilburg, The Netherlands (2001)
2. Bouma, G., Fahmi, I., Mur, J., van Noord, G., van der Plas, L., Tiedeman, J.: Linguistic knowledge and question answering. Traitement Automatique des Langues **2**(46), 15–39 (2005)
3. Bouma, G., Daelemans, W., Hendrickx, I., Hoste, V., Mineur, A.: The COREA-project, manual for the annotation of coreference in Dutch texts. University Groningen (2007)
4. Daelemans, W., van den Bosch, A.: Memory-Based Language Processing. Cambridge University Press, Cambridge, UK/New York, USA (2005)
5. Daelemans, W., Zavrel, J., Berck, P., Gillis, S.: Mbt: a memory-based part of speech tagger generator. In: Proceedings of the 4th ACL/SIGDAT Workshop on Very Large Corpora, pp. 14–27 Santa Cruz, California, USA (1996)
6. Daelemans, W., Hoste, V., De Meulder, F., Naudts, B.: Combined optimization of feature selection and algorithm parameter interaction in machine learning of language. In: Proceedings of the 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia, pp. 84–95 (ECML-2003)
7. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, R., Strassel, S., Weischedel, R.: The automatic content extraction (ACE) program tasks, data, and evaluation. In: Proceedings of the LREC 2004, Lisbon, Portugal, pp. 837–840 (2004)
8. Grishman, R., Sundheim, B.: Coreference task definition. version 2.3. In: Proceedings of the Sixth Message Understanding Conference (MUC-6), Columbia, Maryland, USA pp. 335–344 (1995)

[8] www.tst-centrale.org

9. Hendrickx, I., Hoste, V.: Coreference resolution on blogs and commented news. In: Lalitha Devi, S., Branco, A., Mitkov, R. (eds.) Anaphora Processing and Applications. Lecture Notes in Artificial Intelligence, vol. 5847, pp. 43–53. Springer, Berlin/New York (2009)
10. Hendrickx, I., Hoste, V., Daelemans, W.: Evaluating hybrid versus data-driven coreference resolution. In: Anaphora: Analysis, Algorithms and Application, Proceedings of the DAARC 2007, Lagos, Portugal. Lecture Notes in Artificial Intelligence, vol. 4410, pp. 137–150 (2007)
11. Hendrickx, I., Hoste, V., Daelemans, W.: Semantic and Syntactic Features for Anaphora Resolution for Dutch. Lecture Notes in Computer Science, vol. 4919, pp. 351–361. Springer, Berlin (2008)
12. Hinrichs, E., Kübler, S., Naumann, K.: A unified representation for morphological, syntactic, semantic, and referential annotations. In: Proceedings of the ACL Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, Ann Arbor, MI (2005)
13. Hirschman, L., Robinson, P., Burger, J., Vilain, M.: Automating coreference: the role of annotated training data. In: Proceedings of the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing. Providence, Rhode Island, USA (1997)
14. Hoste, V.: Optimization issues in machine learning of coreference resolution. Ph.D. thesis, Antwerp University, Antwerp, Belgium (2005)
15. Hoste, V., Daelemans, W.: Learning Dutch coreference resolution. In: Proceedings of the Fifteenth Computational Linguistics in The Leiden. The Netherlands (CLIN 2004) (2005)
16. Hoste, V., de Pauw, G.: Knack-2002: a richly annotated corpus of dutch written text. In: Proceedings of LREC 2006, Genoa, Italy, pp. 1432–1437 (2006)
17. Kobdani, H., Schütze, H.: SUCRE: a modular system for coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 92–95. Association for Computational Linguistics, Uppsala, Sweden (2010)
18. Korzen, I., Buch-Kromann, M.: Anaphoric relations in the copenhagen dependency treebanks. In: Beyond Semantics: Corpus-based Investigations of Pragmatic and Discourse Phenomena Proceedings of the DGfS Workshop, Göttingen, Germany pp. 83–98 (2011)
19. Krasavina, O., Chiarcos, C.: PoCoS – Potsdam coreference scheme. In: Proceedings of the Linguistic Annotation Workshop, pp. 156–163. Association for Computational Linguistics, Prague, Czech Republic (2007)
20. Kučová, L., Hajičová, E.: Coreferential relations in the Prague dependency treebank. In: Proceedings of the DAARC 2004, Azores, Portugal, pp. 97–102 (2004)
21. Lendvai, P.: Conceptual taxonomy identification in medical documents. In: Proceedings of the Second International Workshop on Knowledge Discovery and Ontologies, Porto, Portugal pp. 31–38 (2005)
22. Magnini, B., Pianta, E., Girardi, C., Negri, M., Romano, L., Speranza, M., Lenzi, V.B., Sprugnoli, R.: I-CAB: the Italian content annotation bank. In: Proceedings of LREC 2006, Genoa, Italy, pp. 963–968 (2006)
23. MUC-7: Muc-7 coreference task definition. version 3.0. In: Proceedings of the Seventh Message Understanding Conference (MUC-7), Columbia, Maryland, USA (1998)
24. Mur, J.: Increasing the coverage of answer extraction by applying anaphora resolution. In: Fifth Slovenian and First International Language Technologies Conference (IS-LTC '06), Ljubljana, Slovenia (2006)
25. Ng, V., Cardie, C.: Combining sample selection and error-driven pruning for machine learning of coreference rules. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-2002), Philadelphia, PA, USA, pp. 55–62 (2002)
26. Poesio, M.: Discourse annotation and semantic annotation in the GNOME corpus. In: Proceedings of the ACL Workshop on Discourse Annotation, Barcelona, Spain (2004)
27. Poesio, M., Artstein, R.: Anaphoric annotation in the ARRAU corpus. In: Proceedings of the LREC 2008, Marrakech, Morocco, pp. 1170–1174 (2008)
28. Poesio, M., Vieira, R.: A corpus-based investigation of definite description use. Comput. Linguist. **24**(2), 183–216 (1998)
29. Recasens, M., Martí, M.A.: AnCora-CO: coreferentially annotated corpora for Spanish and Catalan. Lang. Res. Eval. **44**(4), 315–345 (2010)

30. Recasens, M., Màrquez, L., Sapena, E., Martí, M.A., Taulé, M., Hoste, V., Poesio, M., Versley, Y.: Semeval-2010 task 1: coreference resolution in multiple languages. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 1–8. Association for Computational Linguistics, Uppsala, Sweden (2010)
31. Rodríguez, K.J., Delogu, F., Versley, Y., Stemle, E.W., Poesio, M.: Anaphoric annotation of wikipedia and blogs in the live memories corpus. In: Proceedings of the LREC 2010. European Language Resources Association (ELRA), Valletta, Malta (2010)
32. Schuurman, I., Hoste, V., Monachesi, P.: Interacting semantic layers of annotation in SoNaR, a reference corpus of contemporary written Dutch. In: Proceedings of the LREC 2010. European Language Resources Association (ELRA), Valletta, Malta (2010)
33. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Comput. Linguist. **27**(4), 521–544 (2001)
34. Tjong Kim Sang, E., Daelemans, W., Höthker, A.: Reduction of Dutch sentences for automatic subtitling. In: Computational Linguistics in The Netherlands 2003. Selected Papers from the Fourteenth CLIN Meeting, Antwerp, Belgium pp. 109–123 (2004)
35. Van de Cruys, T.: Semantic clustering in Dutch. In: Proceedings of the Sixteenth Computational Linguistics in the Netherlands (CLIN), Amsterdam, The Netherlands pp. 17–32 (2005)
36. Vandekerckhove, R.: Belgian Dutch versus Netherlandic Dutch: new patterns of divergence? On pronouns of address and diminutives. Multiling. J. Cross-Cult. Interlang. Commun. **24**, 379–397 (2005)
37. Versley, Y.: Disagreement dissected: vagueness as a source of ambiguity in nominal (co-) reference. In: Ambiguity in Anaphora Workshop Proceedings, pp. 83–89. ESSLLI, Malaga, Spain (2006)
38. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Proceedings of the Sixth Message Understanding Conference (MUC-6), Columbia, Maryland, USA pp. 45–52 (1995)
39. Weischedel, R., Pradhan, S., Ramshaw, L., Palmer, M., Xue, N., Marcus, M., Taylor, A., Greenberg, C., Hovy, E., Belvin, R., Houston, A.: OntoNotes release 3.0. LDC2009T24. Linguistic Data Consortium (2009)
40. Zhekova, D., Kübler, S.: UBIU: a language-independent system for coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 96–99. Association for Computational Linguistics, Uppsala, Sweden (2010)

Chapter 8
Automatic Tree Matching for Analysing Semantic Similarity in Comparable Text

Erwin Marsi and Emiel Krahmer

8.1 Introduction

Natural languages allow us to express essentially the same underlying meaning in a virtually unlimited number of alternative surface forms. In other words, there are often many similar ways to say the same thing. This characteristic poses a problem for natural language processing applications. Automatic summarisers, for example, typically rank sentences according to their informativity and then extract the top n sentences, depending on the required compression ratio. Although the sentences are essentially treated as independent of each other, they typically are not. Extracted sentences may have substantial semantic overlap, resulting in unintended redundancy in the summaries. This is particularly problematic in the case of multi-document summarisation, where sentences extracted from related documents are very likely to express similar information in different ways [21]. Provided semantic similarity between sentences could be detected automatically, this would certainly help to avoid redundancy in summaries.

Similar arguments can be made for many other NLP applications. Automatic duplicate and plagiarism detection beyond obvious string overlap requires recognition of semantic similarity. Automatic question-answering systems may benefit from clustering semantically similar candidate answers. Intelligent document merging software, which supports a minimal but lossless merge of several revisions of the same text, must handle cases of paraphrasing, restructuring, compression, etc. Yet

E. Marsi (✉)
Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
e-mail: emarsi@idi.ntnu.no

E. Krahmer
Department of Communication and Information Sciences, Tilburg University (UvT), Tilburg, The Netherlands
e-mail: e.j.krahmer@uvt.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_8,

another application is in the area of automatic evaluation of machine translation output [20]. The general problem is that even though system output does not superficially match any of the human-produced gold standard translations, it may still be a good translation provided that it expresses the same semantic content. Measuring the semantic similarity between system output and reference translations may therefore be a better alternative to the more superficial evaluation measures currently in use.

In addition to merely *detecting* semantic similarity, we can ask to what extent two expressions share meaning. For instance, the meaning of a sentence can be fully contained in that of another, it may overlap only partly with that of another, etc. This requires an *analysis* of the semantic similarity between a pair of expressions. Like detection, automatic analysis of semantic similarity can play an important role in NLP applications. To return to the case of multi-document summarisation, analysing the semantic similarity between sentences extracted from different documents provides the basis for *sentence fusion*, a process where a new sentence is generated that conveys all common information from both sentences without introducing redundancy [1, 16].

In this paper we present a method for analysing semantic similarity in comparable text. It relies on a combination of morphological and syntactic analysis, lexical resources such as word nets, and machine learning from examples. We propose to analyse semantic similarity between sentences by aligning their syntax trees, where each node is matched to the most similar node in the other tree (if any). In addition, alignments are labeled according to the type of similarity relation that holds between the aligned phrases, which supports further processing. For instance, Marsi and Krahmer [8, 16] describe how to generate different types of sentence fusions on the basis of this relation labelling.

This chapter is structured in the following way. The next section defines the task of matching syntactic trees and labelling alignments in a more formal way. This is followed by an overview of the DAESO corpus, a large parallel monolingual treebank for Dutch, which forms the basis for developing and testing our approach. Section 8.4 outlines an algorithm for simultaneous node alignment and relation labelling. The results of some evaluation experiments are reported in Sect. 8.5. We finish with a discussion of related work and a conclusion.

8.2 Analysing Semantic Similarity

Analysis of semantic similarity can be approached from different angles. A basic approach is to use string similarity measures such as the Levenshtein distance or the Jaccard similarity coefficient. Although cheap and fast, this fails to account for less obvious cases such as synonyms or syntactic paraphrasing. At the other extreme, we can perform a deep semantic analysis of two expressions and rely on formal reasoning to derive a logical relation between them. This approach suffers from issues with coverage and robustness commonly associated with deep

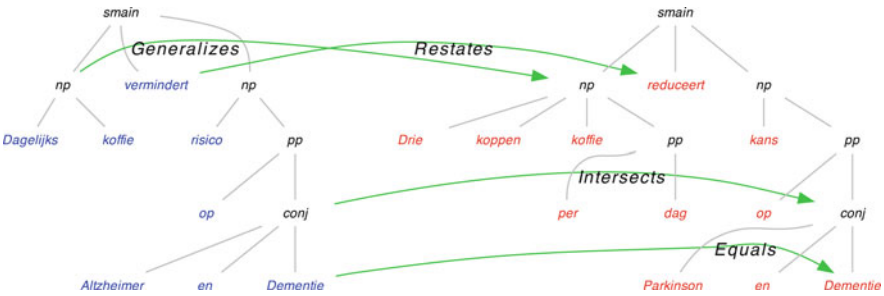


Fig. 8.1 Example of two aligned and labeled syntactic trees. For expository reasons the alignment is not exhaustive

linguistic processing. We therefore argue that the middle ground between these two extremes currently offers the best solution: analysing semantic similarity by means of syntactic tree alignment.

Aligning a pair of similar syntactic trees is the process of pairing those nodes that are most similar. More formally: let v be a node in the syntactic tree T of sentence S and v' a node in the syntactic tree T' of sentence S'. A *labeled node alignment* is a tuple $< v, v', r >$ where r is a label from a set of relations. A *labeled tree alignment* is a set of labeled node alignments. A *labeled tree matching* is a tree alignment in which each node is aligned to at most one other node.

For each node v, its terminal *yield* $\text{STR}(v)$ is defined as the sequence of all terminal nodes reachable from v (i.e., a subsequence of sentence S). Aligning node v to v' with label r indicates that relation r holds between their yields $\text{STR}(v)$ and $\text{STR}(v')$. We label alignments according to a small set of *semantic similarity relations*. As an example, consider the following Dutch sentences:

(1) a. Dagelijks koffie vermindert risico op Alzheimer en Dementie.
Daily coffee diminishes risk of Alzheimer and Dementia.
b. Drie koppen koffie per dag reduceert kans op Parkinson en Dementie.
Three cups coffee a day reduces chance of Parkinson and Dementia.

The corresponding syntax trees and their (partial) alignment is shown in Fig. 8.1. We distinguish the following five mutually exclusive similarity relations:

1. v **equals** v' iff lower-cased $\text{STR}(v)$ and lower-cased $\text{STR}(v')$ are identical – example: *Dementia* equals *Dementia*;
2. v **restates** v' iff $\text{STR}(v)$ is a proper paraphrase of $\text{STR}(v')$ – example: *diminishes* restates *reduces*;
3. v **generalises** v' iff $\text{STR}(v)$ is more general than $\text{STR}(v')$ – example: *daily coffee* generalises *three cups of coffee a day*;
4. v **specifies** v' iff $\text{STR}(v)$ is more specific than $\text{STR}(v')$ – example: *three cups of coffee a day* specifies *daily coffee*;

5. v **intersects** v' iff STR(v) and STR(v') share meaning, but each also contains unique information not expressed in the other – example: *Alzheimer and Dementia* intersects *Parkinson and Dementia*.

Our interpretation of these relations is one of common sense rather than strict logic, akin to the definition of entailment employed in the RTE challenge [4]. Note also that relations are prioritised: *equals* takes precedence over *restates*, etc. Furthermore, *equals*, *restates* and *intersects* are symmetrical, whereas *generalises* is the inverse of *specifies*. Finally, nodes containing unique information, such as *Alzheimer* and *Parkinson*, remain unaligned.

8.3 DAESO Corpus

The DAESO[1] corpus is a parallel monolingual treebank for Dutch that contains parallel and comparable Dutch text from several text domains:

- Alternative Dutch translations of a number of foreign language books
- Auto-cue (text that is automatically presented to a news reader) and subtitle text from news broadcasts by Dutch and Belgium public television channels
- Similar headlines from online news obtained from the Dutch version of Google News
- Similar answers from a Question-Answer corpus in the medical domain
- Press releases about the same news event from two major Dutch press agencies

All text was preprocessed in a number of steps. First, text was obtained by extraction from electronic documents or by OCR and converted to XML. All text material was subsequently processed with a tokeniser for Dutch [22]. OCR and tokenisation errors were in part manually corrected. Next, the Alpino parser for Dutch [2] was used to parse sentences. It provides a relatively theory-neutral syntactic analysis originally developed for the Spoken Dutch Corpus [25]. It is a blend of phrase structure analysis and dependency analysis, with a backbone of phrasal constituents and arcs labeled with syntactic function/dependency labels. Due to time and cost constraints, parsing errors were not subject to manual correction.

The next stage involved aligning similar sentences (regardless of their syntactic structure). This involved automatic alignment using heuristic methods, followed by manual correction using a newly developed alignment annotation tool, called *Hitaext*, for visualising and editing alignments between textual segments.[2] Annotator guidelines specified that aligned sentences must minimally share a "proposition", i.e. a predication over some entity. Just sharing a single entity (typically an noun)

[1] The acronym of the *Detecting And Exploiting Semantic Overlap* research project which gave rise to the corpus; see also http://daeso.uvt.nl

[2] http://daeso.uvt.nl/hitaext

or single predicate (typically a verb or adjective) is insufficient. This prevents alignment of trees which share virtually no content later on.

The final stage consisted of analysing the semantic similarity of aligned sentences along the lines described in the previous section. This included manual alignment of syntactic nodes, as well as labelling these alignments with one of five semantic relations. This work was carried out by six specially trained annotators. For creating and labelling alignments, a special-purpose graphical annotation tool called *Algraeph* was developed.[3]

The resulting corpus comprises over 2.1 M tokens, 678 K of which is manually annotated and 1,511 K is automatically processed. It is freely available for research purposes.[4] It is unique in its size and detailed annotations, and holds great potential for a wide range of research areas.

8.4 Memory-Based Graph Matcher

In order to automatically perform the alignment and labelling tasks described in Sect. 8.2, we cast these tasks simultaneously as a combination of exhaustive pairwise classification using a supervised machine learning algorithm, followed by global optimisation of the alignments using a combinatorial optimisation algorithm. Input to the tree matching algorithm is a pair of syntactic trees consisting of a source tree T_s and a target tree T_t.

Step 1: Feature extraction For each possible pairing of a source node n_s in tree T_s and a target node n_t in tree T_t, create an instance consisting of feature values extracted from the input trees. Features can represent properties of individual nodes, e.g. the category of the source node is NP, or relations between nodes, e.g. source and target node share the same part-of-speech.

Step 2: Classification A generic supervised classifier is used to predict a class label for each instance. The class is either one of the semantic similarity relations or the special class *none*, which is interpreted as *no alignment*. Our implementation employs the memory-based learner TiMBL [3], a freely available, efficient and enhanced implementation of k-nearest neighbour classification. The classifier is trained on instances derived according to Step 1 from a parallel treebank of aligned and labeled syntactic trees.

Step 3: Weighting Associate a cost with each prediction so that high costs indicate low confidence in the predicted class and vice versa. We use the normalised entropy of the class labels in the set of nearest neighbours (H) defined as

[3] http://daeso.uvt.nl/algraeph

[4] www.tst-centrale.org

$$H = -\frac{\sum_{c \in C} p(c) \log_2 p(c)}{\log_2 |C|} \tag{8.1}$$

where C is the set of class labels encountered in the set of nearest neighbours (i.e., a subset of the five relations plus *none*), and $p(c)$ is the probability of class c, which is simply the proportion of instances with class label c in the set of nearest neighbours. Intuitively this means that the cost is 0 if all nearest neighbours are of the same class, whereas the cost goes to 1 if the nearest neighbours are equally distributed over all possible classes.

Step 4: Matching The classification step results in one-to-many alignment of nodes. In order to reduce this to just one-to-one alignments, we search for a node matching which minimises the sum of costs over all alignments. This is a well-known problem in combinatorial optimisation known as the *Assignment Problem*. The equivalent in graph-theoretical terms is a *minimum weighted bipartite graph matching*. This problem can be solved in polynomial time ($O(n^3)$) using e.g., the *Hungarian algorithm* [9]. The output of the algorithm is the labeled tree matching obtained by removing all node alignments labeled with the special *none* relation.

8.5 Experiments

8.5.1 Experimental Setup

These experiments focus on analysing semantic similarity between sentences rather than merely detecting similarity (as a binary classification task). Hence it is assumed that there is at least some semantic overlap between comparable sentences and the task is a detailed analysis of this similarity in terms of a labeled alignment of syntactic constituents.

8.5.1.1 Data Sets

For developing and testing our alignment algorithm, we used half of the manually aligned press releases from the DAESO corpus. This data was divided into a development and held-out test set. The left half of Table 8.1 summarises the respective sizes of development and test set in terms of number of aligned graph pairs, number of aligned node pairs and number of tokens. The percentage of aligned nodes over all graphs is calculated relative to the number of nodes over all graphs. The right half of Table 8.1 gives the distribution of semantic relations in the development and test sets. It can be observed that the distribution is fairly skewed with *equals* being the majority class.

Development was carried out using ten-fold cross validation on the development data and consequently reported scores on the development data are average scores

Table 8.1 Properties of development and test data sets

Data	Graph pairs	Node pairs	Tokens	Aligned nodes (%)	Equals (%)	Restates (%)	Specifies (%)	Generalises (%)	Intersects (%)
Develop	2,664	22,741	45,149	47.20	56.61	6.57	7.52	6.38	22.91
Test	547	4,894	10,005	47.05	58.40	7.11	7.40	6.38	20.72

over ten folds. Only two parameters were optimised on the development set. First, the amount of downsampling of the *none* class was fixed at 20 %; this will be motivated in Sect. 8.5.3. Second, the parameter k of the memory-based classifier – the number of nearest neighbours taken into account during classification – was evaluated in the range from 1 to 15. It was found that $k = 5$ provided the best trade-off between performance and speed. These optimised settings were then applied when testing on the held-out test data.

8.5.1.2 Features

All features used during classification are described in Table 8.2. The word-based features rely on pure string processing and require no linguistic preprocessing. The morphology-based features exploit the limited amount of morphological analysis provided by the Alpino parser [2]. For instance, it provides word roots and decomposes compound words. Likewise the part-of-speech-based features use the coarse-grained part-of-speech tags assigned by the Alpino parser. The lexical-semantic features rely on the Cornetto database [27], an improved and extended version of Dutch WordNet, to look-up synonym and hypernym relations among source and target lemmas. Unfortunately there is no word sense disambiguation module to identify the correct senses, so a certain amount of noise is present in these features. In addition, a background corpus of over 500M words of (mainly) news text provides the word counts required to calculate the Lin similarity measure [11]. The syntax-based features use the syntactic structure, which is a mix of phrase-based and dependency-based analysis. The phrasal features express similarity between the terminal yields of source and target nodes. With the exception of *same-parent-lc-phrase*, these features are only used for full tree alignment, not for word alignment.

We have not yet performed any systematic feature selection experiments. However, we did experiment with a substantial number of other features and combinations. The current feature set resulted from manual tuning on the development set. When removing any of these features, we observed decreased performance.

8.5.1.3 Evaluation Measures

A tree alignment A is a set of node alignments $< v, v' >$ where v and v' are source and target nodes respectively. As sets can be compared using the well-known

Table 8.2 Features[a] used during classification step

Feature	Type	Description
Word		
word-subsumption	string	indicate if source word equals, has as prefix, is a prefix of, has a suffix, is a suffix of, has as infix or is an infix of target word
shared-pre-/in-/suffix-len	int	length of shared prefix/infix/suffix in characters
source/target-stop-word	bool	test if source/target word is in a stop word list
source/target-word-len	int	length of source/target word in characters
word-len-diff	int	word length difference in characters
source/target-word-uniq	bool	test if source/target word is unique in source/target sentence
same-words-lhs/rhs	int	no. of identical preceding/following words in source and target word contexts
Morphology		
root-subsumption	string	indicate if source root equals, has as prefix, is a prefix of, has a suffix, is a suffix of, has as infix or is an infix of target root
roots-share-pre-/in-/suffix	bool	source and target root share a prefix/infix/suffix
Part-of-speech		
source/target-pos	string	source/target part-of-speech
same-pos	bool	test if source and target have same part-of-speech
source/target-content	bool	test if source/target word is a content word
both-content-word	bool	test if both source and target word are content words
Lexical-semantic using Cornetto		
cornet-restates	float	1.0 if source and target words are synonyms and 0.5 if they are near-synonyms, zero otherwise
cornet-specifies	float	Lin similarity score if source word is a hyponym of target word
cornet-generalises	float	Lin similarity score if source word is a hypernym of target word
cornet-intersects	float	Lin similarity score if source word share a common hypernym
Syntax		
source/target-cat	string	source/target syntactic category
same-cat	bool	test if source and target have same syntactic category
source/target-parent-cat	string	source/target syntactic category of parent node
same-parent-cat	bool	test if parents of source and target have same syntactic category
source/target-deprel	string	source/target dependency relation
same-deprel	bool	test if source and target have same dependency relation
same-dephead-root	bool	test if the dependency heads of source and target have same root
Phrasal		
word-prec/rec	float	precision/recall on the yields of source and target nodes
same-lc-phrase	bool	test if lower-cased yields of source and target nodes are identical
same-parent-lc-phrase	bool	test if lower-cased yields of parents of nodes are identical
source/target-phrase-len	int	length of source/target phrase in words
phrase-len-diff	int	phrase length difference in words

[a]slashes indicate multiple versions of the same feature, e.g. *source/target-pos* represents the two features *source-pos* and *target-pos*

precision and *recall* measures [26], the same measures can be applied to alignments. Given that A_{true} is a true tree alignment and A_{pred} is a predicted tree alignment, precision and recall are defined as follows:

$$precision = \frac{|A_{true} \cap A_{pred}|}{|A_{pred}|} \tag{8.2}$$

$$recall = \frac{|A_{true} \cap A_{pred}|}{|A_{true}|} \tag{8.3}$$

Precision and recall are combined in the F_1 score, which is defined as the harmonic mean between the two, giving equal weight to both terms, i.e. $F_1 score = (2 * precision * recall)/(precision + recall)$

The same measures can be used for comparing *labeled* tree alignments in a straight forward way. Recall that a labeled tree alignment is a set of labeled node alignments $< v, v', r >$ where v is a source node, v' a target node and r is a label from the set of semantic similarity relations. Let A^{rel} be the subset of all alignments in A with label *rel*, i.e. $A^{rel} = \{< v_s, v_t, r > \in A : r = rel\}$. This allows us to calculate, for example, precision on relation *equals* as follows.

$$precision^{EQ} = \frac{|A^{EQ}_{true} \cap A^{EQ}_{pred}|}{|A^{EQ}_{pred}|} \tag{8.4}$$

We thus calculate precision as in the unlabelled case, but ignore all alignments – whether true or predicted – labeled with a different relation. Recall and F score on a particular relation can be calculated in a similar fashion.

8.5.2 Results on Tree Alignment

Table 8.3 presents the results on tree alignment consisting of baseline, human and MBGM scores.

8.5.2.1 Baseline Scores

A simple greedy alignment procedure served as baseline. For word alignment, identical words are aligned as *equals* and identical roots as *restates*. For full tree alignment, this is extended to the level of phrases so that phrases with identical words are aligned as *equals* and phrases with identical roots as *restates*. The baseline does not predict *specifies*, *generalises* or *intersects* relations, as that would require a more involved, knowledge-based approach, relying on resources such as a wordnet.

8.5.2.2 Human Scores

A subset of the test data, consisting of 10 similar press releases comprising a total of 48 sentence pairs, was independently annotated by 6 annotators to determine

Table 8.3 Scores (in percentages) on tree alignment and semantic relation labelling

			Labelling:						
	Alignment:		Eq:	Re:	Spec:	Gen:	Int:	Macro:	Micro:
	Prec:	82.50	83.76	46.72	0.00	0.00	0.00	26.10	82.18
Develop baseline:	Rec:	54.54	93.66	20.01	0.00	0.00	0.00	22.74	54.34
	F:	65.67	88.43	28.02	0.00	0.00	0.00	23.29	65.42
	Prec:	86.40	95.08	45.22	41.45	44.95	64.17	58.18	78.66
Develop MBGM:	Rec:	86.06	96.16	35.86	31.16	39.06	72.21	54.89	78.35
	F:	86.23	95.62	40.00	35.58	41.80	67.95	56.19	78.51
	Prec:	84.23	85.68	42.24	0.00	0.00	0.00	25.58	84.14
Test baseline:	Rec:	56.21	94.44	14.08	0.00	0.00	0.00	21.70	56.15
	F:	67.43	89.85	21.12	0.00	0.00	0.00	22.19	67.35
	Prec:	86.87	95.96	51.79	40.43	38.36	60.87	57.48	78.10
Test MBGM:	Rec:	86.46	96.27	40.56	32.20	34.23	70.35	54.72	77.88
	F:	86.66	96.11	45.49	35.85	36.18	65.27	55.78	77.99
Human:	F:	88.31	95.83	71.38	60.21	66.71	62.67	71.36	81.92

inter-annotator agreement on the alignment and labelling tasks. Given the six annotations $A_1, \ldots, A_6$, we repeatedly took one as the A_{true} against which the five other annotations were evaluated as A_{pred}. We then computed the average scores over these $6 * 5 = 30$ scores.[5] This resulted in an F-score of 88.31 % on alignment only. For relation labelling, the scores differed per relation, as is to be expected: the average F-score for *equals* was 95.83 % alignment,[6] and for the other relations average F-scores between 62 and 72 % were obtained.

8.5.2.3 System Scores

The first thing to observe is that the MBGM scores on the development and tests sets are very similar throughout, suggesting that generalisation across the news domain is fairly good. We will therefore focus on the test scores, comparing them statistically with the baseline scores and informally with the human scores.

With an alignment F-score on the test set of 86.66 %, MBGM scores over 19 % higher than the baseline system, which is significant ($t(18) = 25.68$, $p < 0.0001$). This gain is mainly due to a much better recall score. This F-score is also less than

[5] As a result of this procedure, precision, recall and F score end up being equal.

[6] At first sight, it may seem that labelling *equals* is a trivial and deterministic task, for which the F-score should always be close to 100 %. However, the same word may occur multiple times in the source or target sentences, which introduces ambiguity. This frequently occurs with function words such as determiners and prepositions. Moreover, choosing among several equivalent *equals* alignments may sometimes involve a somewhat arbitrary decision. This situation arises, for instance, when a proper noun is mentioned just once in the source sentence but twice in the target sentence.

2 % lower than the average alignment F-score obtained by our human annotators, albeit on a subset of test data.

In a similar vein, the performance of MBGM on relation labelling is considerably better than that of the baseline system, significantly outperforming the baseline for each semantic relation ($t(18) > 12.6636, p < 0.0001$), trivially so for the *specifies*, *generalises specifies* and *intersects* relations, which the baseline system never predicts.

The macro scores are plain averages over the five scores on each relation, whereas the micro scores are weighted averages. As *equals* is the majority class and at the same time easiest to predict, the micro scores are higher. The macro scores, however, better reflect performance on the real challenge, that is, correctly predicting the relations other than *equals*. MBGM scores a macro F-score of 55.78 % (an improvement of over 33 % over the baseline) and a micro average of 77.99 % (over 10 % above the baseline). It is interesting to observe that MBGM obtains *higher* F-scores on *equals* and *intersects* (the two most frequent relations) than the human annotators. As a result of this, the micro F-score of the automatic tree alignment is merely 4 % lower than the human reference counterpart. However, MBGM's macro F-score (55.78) is still well below the human score (71.36).

8.5.3 Effects of Downsampling

As described in Sect. 8.4, MBGM performs tree alignment by initially considering every possible alignment from source nodes to target nodes. For each possible pairing of a source node n_s in tree T_s and a target node n_t in tree T_t, an instance is created consisting of feature values extracted from the input trees. A memory-based classifier is then used to predict a class label for each instance, either one of the semantic similarity relations or the special class *none*, which is interpreted as *no alignment*. The vast majority of the training instances is of class *none*, because a node is aligned to at most one node in the other tree and unaligned to all other nodes in the same tree. The class distribution in the development data is: *equals* 0.81 %, *restates* 0.08 %, *specifies* 0.07 %, *generalises* 0.10 %, *intersects* 0.31 %, *none* 98.63 %. The problem is that most classifiers have difficulties with handling heavily skewed class distributions, usually causing them to always predict the majority class. We address this by downsampling the *none* class (in the training data) so that less frequent classes become more likely to be predicted.

The effects of downsampling are shown in Fig. 8.2 where precision, recall and F-score are plotted as a function of the percentage of original *none* instances in the training data. The training and test material correspond to a 90/10 % split of the development data. Timbl was used with its default settings, except for $k = 5$. The first plot shows scores on alignment regardless of relation labelling. The general trend is that downsampling increases the recall at the cost of precision until a cross-over point at around 20 %. This effect is mainly due to the fact that downsampling increases the number of predictions other than *none*.

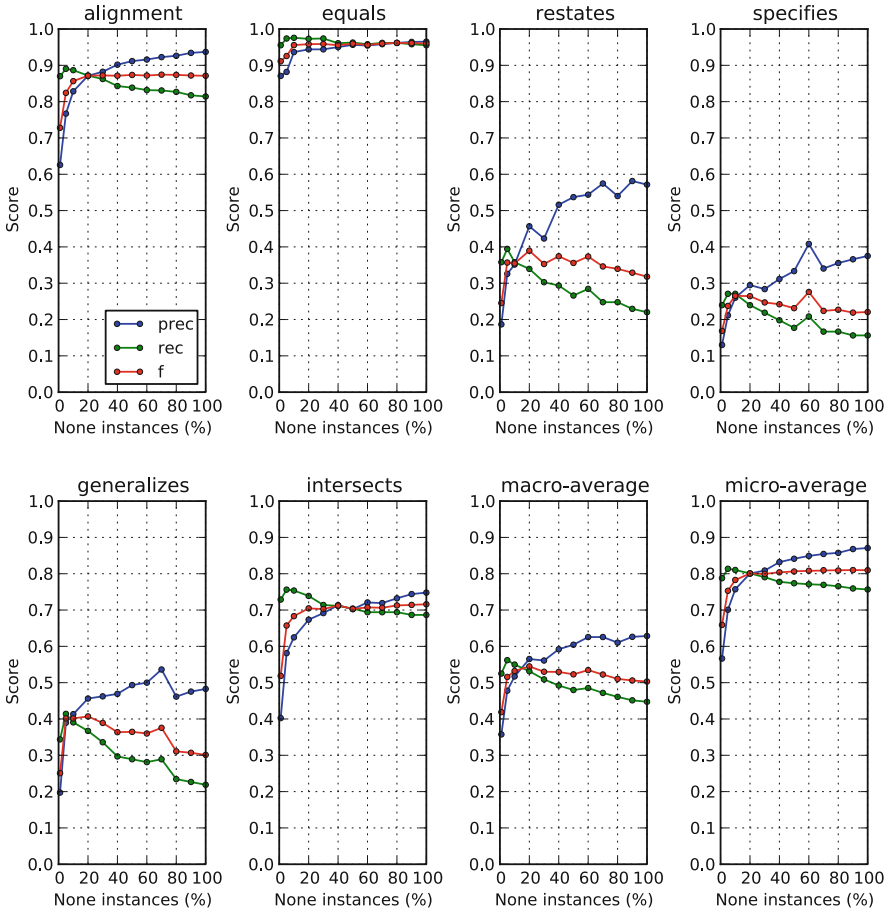


Fig. 8.2 Effects of downsampling *none* instances with regard to precision, recall and F-score, first for alignment only (i.e. ignoring relation label), next per alignment relation and finally as macro/micro average over all relations

The next five plots show the effect of downsampling per alignment relation. The cross-over point is higher for *equals* and *intersects*, at about 40 %. As these are still relatively frequent relations, their F-score is not negatively affected by all the *none* instances. However, for the least frequent relations – *restates*, *specifies*, *generalises* – it can be observed that the F-score is going down when using more than 20 % of the *none* instances. A pattern that is reflected in the macro-average plot (i.e. plain average score over all five relations), while the micro-average plot (i.e. weighted average) is more similar to those for *equals* and *intersects*, as it is dominated by these two most frequent relations.

Even though the alignment only and micro-average F-scores are marginally best without any downsampling, we choose to report results with downsampling *none* to

20 %, because this yields the optimal macro-average F-score. Arguably the optimal downsampling percentage may be specific to the data set and may change with, for example, more training data or another value of the k parameter in nearest neighbour classification.

8.5.4 Effects of Training Data Size

To study the effects of more training data on the scores, experiments were run gradually increasing the amount of training data from 1 up to 100 %. The experimental setting was the same as described in the previous section, including a constant downsampling to 20 % of the *none* class. The resulting learning curves are shown in Fig. 8.3. The learning curve for alignment only suggests that the learner is saturated at about 50 % of the training data, after which precision and recall are virtually identical and the F-score improves only very slowly. With regard to the alignment relations, *equals* and *intersects* show similar behaviour, with arguably no gain in performance after using more than half of the training data. Being dominated by these two relations, the same goes for the micro average scores. For *restates* and *generalises*, however, we find that scores are getting better, and further improvement may therefore be expected with even more training data. The only outlier is *specifies*, with scores that appear to go down somewhat when more training data is consumed. Until further study, we consider this an artefact of the test data. The general trend that the learner is not yet saturated with training samples for the less frequent relations is also reflected in the still improving macro-average scores.

8.6 Related Work

Many syntax-based approaches to machine translation rely on bilingual treebanks to extract transfer rules or train statistical translation models. In order to build bilingual treebanks a number of methods for automatic tree alignment have been developed, e.g., [5, 6, 10, 24]. Most related to our approach is the work on discriminative tree alignment by Tiedemann and Kotzé [23]. However, these algorithms assume that source and target sentences express the same information (i.e. *parallel* text) and cannot cope with comparable text where parts may remain unaligned. See [12] for further arguments and empirical evidence that MT alignment algorithms are not suitable for aligning parallel monolingual text.

Recognising textual entailments (RTE) could arguably be seen as a specific instance of detecting semantic similarity [4]. The RTE task is commonly defined as: given a text T (usually consisting of one or two sentences) determine whether a sentence H (the hypothesis) is entailed by T. Various researchers have attempted to use alignments between T and H to predict textual entailments [7, 18]. However, these RTE systems have a directional bias (i.e., they assume the text is longer

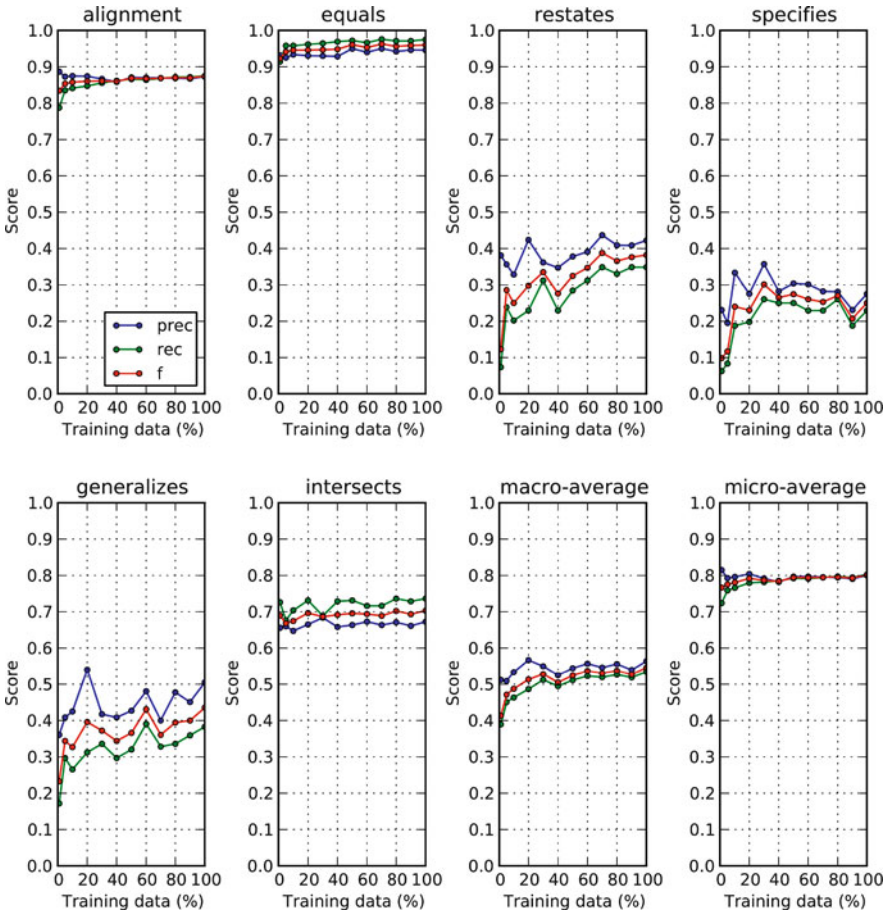


Fig. 8.3 Effects of training data size on precision, recall and F-scores, first for alignment only (i.e. ignoring relation label), next per alignment relation and finally as macro/micro average over all relations

than the the hypothesis), and apart from an entailment judgement do not provide an analysis of semantic similarity. Our *specifies* relation may be interpreted as entailment and vice versa, our *generalises* relation as reversed entailment. Likewise, *restates* may be regarded as mutual entailment. The *intersects* relation, however, cannot be stated in terms of entailment, which makes our relations somewhat more expressive. For instance, it can express the partial similarity in meaning between "*John likes milk*" and "*John likes movies*". In a similar way, contradictory statements such as "*John likes milk*" versus "*John hates milk*" can not be distinguished from completely unrelated statements such as "*John likes milk*" and "*Ice is cold*" in terms of entailment. In contrast, *intersects* is capable of capturing the partial similarity between contradictory statements.

Marneffe et al. [14] align semantic graphs for textual inference in machine reading, both manually and automatically. Although they do use typed dependency graphs, the alignment is only at the token level, and no explicit phrase alignment is carried out. As part of manual annotation, alignments are labeled with relations akin to ours (e.g. 'directional' versus 'bi-directional'), but their automatic alignment does not include labelling. MacCartney, Galley, and Manning [12] describe a system for monolingual phrase alignment based on supervised learning which also exploits external resources for knowledge of semantic relatedness. In contrast to our work, they do not use syntactic trees or similarity relation labels. Partly similar semantic relations are used in [13] for modelling semantic containment and exclusion in natural language inference. Marsi and Krahmer [15] is closely related to our work, but follows a more complicated method: first a dynamic programming-based tree alignment algorithm is applied, followed by a classification of similarity relations using a supervised-classifier. Other differences are that their data set is much smaller and consists of parallel rather than comparable text. A major drawback of this algorithmic approach is that it cannot cope with crossing alignments, which occur frequently in the manually aligned DAESO corpus. We are not aware of other work that combines alignment with semantic relation labelling, or algorithms which perform both tasks simultaneously.

8.7 Conclusions

We have proposed to analyse semantic similarity between comparable sentences by aligning their syntax trees, matching each node to the most similar node in the other tree (if any). In addition, alignments are labeled with a semantic similarity relation. We have reviewed the DAESO corpus, a parallel monolingual treebank for Dutch consisting of over two million tokens and covering both parallel and comparable text genres. It provides detailed analyses of semantically similar sentences in the form of syntactic node alignments and alignment relation labelling. We have subsequently presented a Memory-based Graph Matcher (MBGM) that performs both of these tasks simultaneously as a combination of exhaustive pairwise classification using a memory-based learning algorithm, and global optimisation of alignments using a combinatorial optimisation algorithm. It relies on a combination of morphological/syntactic analysis, lexical resources such as word nets, and machine learning using a parallel monolingual treebank. Results on aligning comparable news texts from the DAESO corpus show that MBGM consistently and significantly outperforms the baseline, both for alignment and labelling.

In future research we will test MBGM on other data, as the DAESO corpus contains other segments with various degrees of semantic overlap. We also intend to explore additional features which facilitate learning of lexical and syntactic paraphrasing patterns, for example, vector space models for word similarity. In addition, a comparison with other alignment systems, such as Giza++ [19], would provide a stronger baseline.

Acknowledgements This work was conducted within the DAESO project with participation from Tilburg University (Emiel Krahmer and Erwin Marsi), Antwerp University (Walter Daelemans and Iris Hendrickx), University of Amsterdam (Maarten de Rijke and Edgar Meij) and Textkernel (Jakub Zavrel and Martijn Spitters). We would also like to thank our three anonymous reviewers for their constructive criticism. Some parts of this work have been published before, most notably in [17].

References

1. Barzilay, R., McKeown, K.R.: Sentence fusion for multidocument news summarization. Comput. Linguist **31**(3), 297–328 (2005)
2. Bouma, G., van Noord, G., Malouf, R.: Alpino: Wide-coverage computational analysis of Dutch. In: Daelemans, W., Sima'an, K., Veenstra, J., Zavre, J. (eds.) Computational Linguistics in the Netherlands 2000, pp. 45–59. Rodopi, Amsterdam/New York (2001)
3. Daelemans, W., Zavrel, J., Van der Sloot, K., Van den Bosch, A.: TiMBL: Tilburg Memory Based Learner, version 6.2, reference manual. Tech. Rep. ILK 09-01, Induction of Linguistic Knowledge, Tilburg University (2009)
4. Dagan, I., Glickman, O., Magnini, B.: The PASCAL recognising textual entailment challenge. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK (2005)
5. Gildea, D.: Loosely tree-based alignment for machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Sapporo, Japan, pp. 80–87 (2003)
6. Groves, D., Hearne, M., Way, A.: Robust sub-sentential alignment of phrase-structure trees. In: Proceedings of the 20th International Conference on Computational Linguistics (CoLing '04), Geneva, Switzerland, pp. 1072–1078 (2004)
7. Herrera, J., nas, A.P., Verdejo, F.: Textual entailment recognition based on dependency analysis and wordnet. In: Proceedings of the 1st PASCAL Recognision Textual Entailment Challenge Workshop, Southampton, UK (2005)
8. Krahmer, E., Marsi, E., van Pelt, P.: Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. In: Moore, J., Teufel, S., Allan, J., Furui, S. (eds.) Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Columbus, Ohio, USA, pp. 193–196 (2008)
9. Kuhn, H.W.: The Hungarian Method for the assignment problem. Nav. Res. Logist. Q. **2**, 83–97 (1955)
10. Lavie, A., Parlikar, A., Ambati, V.: Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In: Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation, Columbus, Ohio, USA, pp. 87–95 (2008)
11. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, Madison, Wisconsin, USA, pp. 296–304 (1998)
12. MacCartney, B., Galley, M., Manning, C.D.: A phrase-based alignment model for natural language inference. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, pp. 802–811 (2008)

13. MacCartney, B., Manning, C.: Modeling semantic containment and exclusion in natural language inference. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, Manchester, UK, pp. 521–528 (2008)
14. de Marneffe, M., Grenager, T., MacCartney, B., Cer, D., Ramage, D., Kiddon, C., Manning, C.: Aligning semantic graphs for textual inference and machine reading. In: Proceedings of the AAAI Spring Symposium, Stanford, USA (2007)
15. Marsi, E., Krahmer, E.: Classification of semantic relations by humans and machines. In: Proceedings of the ACL 2005 Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor, Michigan, pp. 1–6 (2005)
16. Marsi, E., Krahmer, E.: Explorations in sentence fusion. In: Proceedings of the 10th European Workshop on Natural Language Generation, Aberdeen, GB (2005)
17. Marsi, E., Krahmer, E.: Automatic analysis of semantic similarity in comparable text through syntactic tree matching. In: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pp. 752–760. Coling 2010 Organizing Committee, Beijing, China (2010)
18. Marsi, E., Krahmer, E., Bosma, W., Theune, M.: Normalized alignment of dependency trees for detecting textual entailment. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy, pp. 56–61 (2006)
19. Och, F.J., Ney, H.: Improved statistical alignment models. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL '00, pp. 440–447. Association for Computational Linguistics, Stroudsburg, PA, USA (2000)
20. Padó, S., Cer, D., Galley, M., Jurafsky, D., Manning, C.: Measuring machine translation quality as semantic equivalence: a metric based on entailment features. Mach. Transl. **23**, 181–193 (2009)
21. Radev, D., McKeown, K.: Generating natural language summaries from multiple on-line sources. Comput. Linguist. **24**(3), 469–500 (1998)
22. Reynaert, M.: Sentence-splitting and tokenization in d-coi. Tech. Rep. 07-07, ILK Research Group (2007)
23. Tiedemann, J., Kotzé, G.: Building a large machine-aligned parallel treebank. In: Eighth International Workshop on Treebanks and Linguistic Theories, Milan, Italy, p. 197 (2009)
24. Tinsley, J., Zhechev, V., Hearne, M., Way, A.: Robust language-pair independent sub-tree alignment. Mach. Transl. Summit **XI**, 467–474 (2007)
25. van der Wouden, T., Hoekstra, H., Moortgat, M., Renmans, B., Schuurman, I.: Syntactic analysis in the spoken dutch corpus. In: Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas, Canary Islands, Spain, pp. 768–773 (2002)
26. van Rijsbergen, C.: Information Retrieval, 2nd edn. Butterworth, London/Boston (1979)
27. Vossen, P., Maks, I., Segers, R., van der Vliet, H.: Integrating lexical units, synsets and ontology in the Cornetto Database. In: Proceedings of the LREC 2008, Marrakech, Morocco (2008)

Chapter 9
Large Scale Syntactic Annotation of Written Dutch: Lassy

Gertjan van Noord, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste

9.1 Introduction

The construction of a 500-million-word reference corpus of written Dutch has been identified as one of the priorities in the STEVIN programme. The focus is on written language in order to complement the Spoken Dutch Corpus (CGN) [13], completed in 2003. In D-COI (a pilot project funded by STEVIN), a 50-million-word pilot corpus has been compiled, parts of which were enriched with verified syntactic annotations. In particular, syntactic annotation of a sub-corpus of 200,000 words has been completed. Further details of the D-COI project can be found in Chap. 13, p. 219. In Lassy, the sub-corpus with verified syntactic annotations has been extended to one million words. We refer to this sub-corpus as Lassy Small. In addition, a much larger corpus has been annotated with syntactic annotations automatically. This larger corpus is called Lassy Large. Lassy Small contains corpora compiled in STEVIN D-COI, some corpora from STEVIN DPC (cf. Chap. 11, p. 185), and some excerpts from the Dutch Wikipedia. Lassy Large includes the corpora compiled in the STEVIN SONAR project [7]—cf. Chap. 13, p. 219.

The Lassy project has extended the available syntactically annotated corpora for Dutch both in size as well as with respect to the various text genres and topical domains. In order to judge the quality of the resources, the annotated corpora have

G. van Noord (✉) · G. Bouma · D. de Kok · J. van der Linde · E. Tjong Kim Sang
University of Groningen, Groningen, The Netherlands
e-mail: g.j.m.van.noord@rug.nl; g.bouma@rug.nl; d.j.a.de.kok@rug.nl; jelmer@ikhoefgeen.nl; e.f.tjong.kim.sang@rug.nl

F. Van Eynde · I. Schuurman · V. Vandeghinste
KU Leuven, Leuven, Belgium
e-mail: frank@ccl.kuleuven.be; ineke.schuurman@ccl.kuleuven.be; vincent@ccl.kuleuven.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_9,

been externally validated by the Center for Sprogteknologi of the University of Copenhagen. In Sect. 9.5 we present the results of this external validation.

In addition, various browse and search tools for syntactically annotated corpora have been developed and made freely available. Their potential for applications in corpus linguistics and information extraction has been illustrated and evaluated through a series of three case studies.

In this article, we illustrate the potential of the Lassy treebanks by providing a short introduction to the annotations and the available tools, and by describing a number of typical research cases which employ the Lassy treebanks in various ways.

9.2 Annotation and Representation

In this section we describe the annotations in terms of part-of-speech, lemma and syntactic dependency. Furthermore, we illustrate how the annotations are stored in a straightforward XML file format.

Annotations for Lassy Small have been assigned in a semi-automatic manner. Annotations were first assigned automatically, and then our annotators manually inspected these annotations and corrected the mistakes. For part-of-speech and lemma annotation, Tadpole [12] has been used. For the syntactic dependency annotation, we used the Alpino parser [15]. For the correction of syntactic dependency annotations, we employed the TrEd tree editor [8]. In addition, a large number of heuristics have been applied to find annotation mistakes semi-automatically.

The annotations for Lassy Large have been assigned in the same way, except that there has been no manual inspection and correction phase.

9.2.1 Part-of-Speech and Lemma Annotation

The annotations include part-of-speech annotation and lemma annotation for words, and syntactic dependency annotation between words and word groups. Part-of-speech and lemma annotation closely follow the guide-lines developed in D-COI [14]. These guide-lines extend the guide-lines developed in CGN, in order to take into account typical constructs for written language. As an example of the annotations, consider the sentence

(1) In 2005 moest hij stoppen na een meningsverschil met de studio .
In 2005 must he stop after a dispute with the studio .
In 2005, he had to stop after a dispute with the studio

The annotation of this sentence is given here as follows where each line contains the word, followed by the lemma, followed by the part-of-speech tag.

```
In              in              VZ(init)
2005            2005            TW(hoofd,vrij)
```

```
moest            moeten           WW(pv,verl,ev)
hij              hij              VNW(pers,pron,nomin,vol,3,ev,masc)
stoppen          stoppen          WW(inf,vrij,zonder)
na               na               VZ(init)
een              een              LID(onbep,stan,agr)
meningsverschil  meningsverschil  N(soort,ev,basis,onz,stan)
met              met              VZ(init)
de               de               LID(bep,stan,rest)
studio           studio           N(soort,ev,basis,zijd,stan)
.                .                LET()
```

As the example indicates, the annotation not only provides the main part-of-speeches such as VZ (preposition), WW (verb), VNW (pronoun), but also various features to indicate tense, aspect, person, number, gender etc. Below, we describe how the part-of-speech and lemma annotations are included in the XML representation, together with the syntactic dependency annotations.

9.2.2 Syntactic Dependency Annotation

The guide-lines for the syntactic dependency annotation are given in detail in [18]. This manual is a descendent of the CGN and D-Coi syntactic annotation manuals. The CGN syntactic annotation manual [5] has been extended with many more examples and adapted by reducing the amount of linguistic discussions. Some further changes were applied based on the feedback in the validation report of the D-COI project. In a number of documented cases, the annotation guidelines themselves have changed in order to ensure consistency of the annotations, and to facilitate semi-automatic annotation.

Syntactic dependency annotations express three types of syntactic information:

- hierarchical information: which words belong together
- relational information: what is the grammatical function of the various words and word groups (such functions include head, subject, direct object, modifier etc.)
- categorial information: what is the categorial status of the various word groups (categories include NP, PP, SMAIN, SSUB, etc.)

As an example of a syntactic dependency annotation, consider the graph in Fig. 9.1 for the example sentence given above. This example illustrates a number of properties of the representation. Firstly, the left-to-right surface order of the word is not always directly represented in the tree structure, but rather each of the leaf nodes is associated with the position in the string that it occurred at (the subscript). The tree structure does represent hierarchical information, in that words that belong together are represented under one node. Secondly, some word groups are analysed to belong to more than a single word group. We use a co-indexing mechanism to represent such *secondary edges*. In this example, the word `hij` functions both as the subject of the modal verb `moeten` and the main verb `stoppen`—this is indicated by the index 1. Thirdly, we inherit from CGN the practice that punctuation (including sentence internal punctuation) is not analysed syntactically, but simply attached

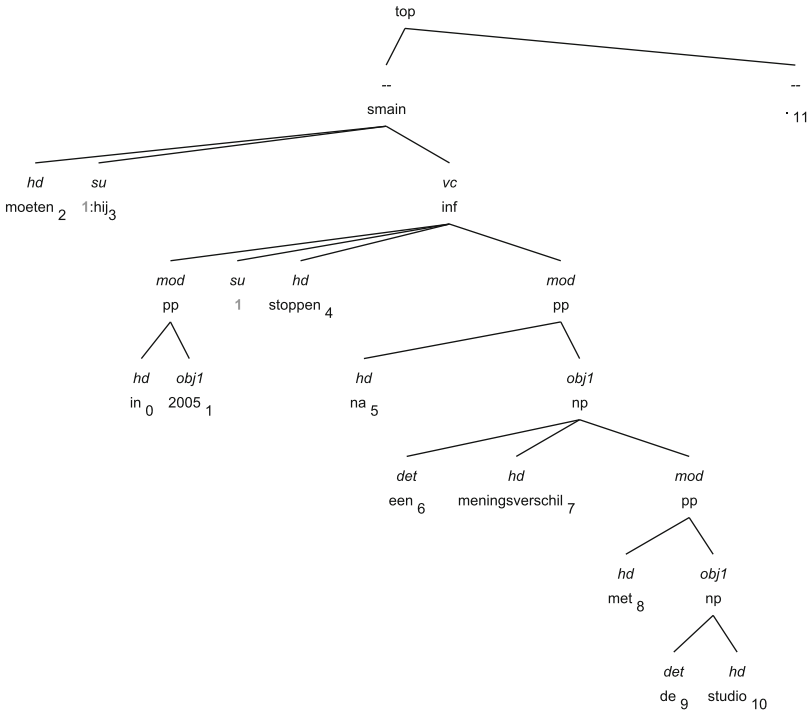


Fig. 9.1 Syntactic dependency graph for sentence (1)

to the top node, to ensure that all tokens of the input are part of the dependency structure. The precise location of punctuation tokens is represented because all tokens are associated with an integer indicating the position in the sentence.

9.2.3 Representation in XML

Both the dependency structures and the part-of-speech and lemma annotations are stored in a single XML format. Advantages of the use of XML include the availability of general purpose search and visualisation software. For instance, we exploit XPath[1] (standard XML query language) to search in large sets of dependency structures, and XQuery to extract information from such large sets of dependency structures.

In the XML-structure, every node is represented by a `node` entity. The other information is presented as values of various XML-attributes of those nodes.

[1] http://www.w3.org/TR/xpath/ and http://www.w3.org/TR/xpath20/

The important attributes are `cat` (syntactic category), `rel` (grammatical function), `postag` (part-of-speech tag), `word`, `lemma`, `index`, `begin` (starting position in the surface string) and `end` (end position in the surface string).

Ignoring some attributes for expository purposes, part of the annotation of our running example is given in XML as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<alpino_ds version="1.3">
  <node begin="0" cat="top" end="12" rel="top">
    <node begin="0" cat="smain" end="11" rel="--">
      <node begin="2" end="3" lemma="moeten" word="moest"/>
      <node begin="3" end="4" index="1" lemma="hij"
            rel="su" word="hij"/>
      <node begin="0" cat="inf" end="11" rel="vc">
        <node begin="0" cat="pp" end="2" rel="mod">
          <node begin="0" end="1" lemma="in"
                rel="hd" word="In"/>
          <node begin="1" end="2" lemma="2005"
                rel="obj1" word="2005"/>
        </node>
        <node begin="3" end="4" index="1" rel="su"/>
        ....
      </node>
    </node>
    <node begin="11" end="12" lemma="." rel="--" word="."/>
  </node>
  <sentence>In 2005 moest hij stoppen na een meningsverschil
             met de studio .</sentence>
</alpino_ds>
```

Leaf nodes have further attributes to represent the part-of-speech tag. The attribute `postag` will be the part-of-speech tag including the various sub-features. The abbreviated part-of-speech tag—the part without the attributes—is available as the value of the attribute `pt`. In addition, each of the sub-features itself is available as the value of further XML attributes. The precise mapping of part-of-speech tags and attributes with values is given in [18]. The actual node for the finite verb `moest` in the example including the attributes to represent the part-of-speech is:

```
<node begin="2" end="3" lemma="moeten" postag="WW(pv,verl,ev)"
      pt="ww" pvagr="ev" pvtijd="verl" rel="hd" word="moest"
      wvorm="pv"/>
```

This somewhat redundant specification of the information encoded in the part-of-speech labels facilitates the construction of queries, since it is possible to refer directly to particular sub-features, and therefore to generalise more easily over part-of-speech labels.

9.3 Querying the Treebanks

As the annotations are represented in XML, there is a variety of tools available to work with the annotations. Such tools include XSLT, XPath and XQuery, as well as a number of special purpose tools—some of which were developed in the course of the Lassy project. Most of these tools have in common that particular parts of the

tree can be identified using the XPath query language. XPath (XML Path Language) is an official W3C standard which provides a language for addressing parts of an XML document. In this section we provide a number of simple examples of the use of XPath to search in the Lassy corpora. We then continue to argue against some perceived limitations of XPath.

9.3.1 Search with XPath

We start by providing a number of simple XPath queries that can be used to search in the Lassy treebanks. We do not give a full introduction to the XPath language—for this purpose there are various resources available on the web.

9.3.1.1 Some Examples

With XPath, we can refer to hierarchical information (encoded by the hierarchical embedding of node elements), grammatical categories and functions (encoded by the `cat` and `rel` attributes), and surface order (encoded by the attributes `begin` and `end`.

As a simple introductory example, the following query:

```
//node[@cat="pp"]
```

identifies all nodes anywhere in a given document, for which the value of the `cat` attribute equals `pp`. In practice, if we use such a query against our Lassy Small corpus using the Dact tool (introduced below), we will get all sentences which contain a prepositional phrase. In addition, these prepositional phrases will be highlighted. In the query we use the double slash notation to indicate that this `node` can appear anywhere in the dependency structure. Conditions about this node can be given between square brackets. Such conditions often refer to particular values of particular attributes. Conditions can be combined using the boolean operators `and`, `or` and `not`. For instance, we can extend the previous query by requiring that the `PP` node should start at the beginning of the sentence:

```
//node[@cat="pp" and @begin="0"]
```

Brackets can be used to indicate the intended structure of the conditions, as in:

```
//node[(@cat="pp" or @cat="advp") and not(@begin="0")]
```

Conditions can also refer to the context of the node. In the following query, we pose further restrictions on a daughter node of the PP category.

```
//node[@cat="pp" and node[@rel="hd" and not(@pt="vz")]]
```

This query will find all sentences in which a PP occurs with a head node for which it is the case that its part-of-speech label is not of the form `VZ(..)`. Such a query will

return quite a few hits—in most cases for prepositional phrases which are headed by multi-word-units such as `in tegenstelling tot` (in contrast with), `met betrekking tot` (with respect to), ... If we want to exclude such multi-word-units, the query could be extended as follows, where we require that there is a `word` attribute, irrespective of its value.

```
//node[@cat="pp" and
       node[@rel="hd" and @word and not(@pt="vz")]]
```

We can look further down inside a node using the single slash notation. For instance, the expression `node[@rel="obj1"]/node[@rel="hd"]` will refer to the head of the direct object. We can also access the value of an attribute of a sub-node as in `node[@rel="hd"]/@postag`.

It is also possible to refer to the mother node of a given node, using the double dot notation. The following query identifies prepositional phrases which are a dependent in a main sentence:

```
//node[@cat="pp" and ../@cat="smain"]
```

Combining the two possibilities we can also refer to sister nodes. In this query, we find prepositional phrases as long as there is a sister which functions as a secondary object:

```
//node[@cat="pp" and ../node[@rel="obj2"]]
```

Finally, the special notation `.//` identifies any node which is embedded anywhere in the current node. The next query finds embedded sentences which include the word `van` anywhere.

```
//node[@cat="ssub" and .//node[@word="van"]]
```

9.3.1.2 Left to Right Ordering

Consider the following example, in which we identify prepositional phrases in which the preposition (the head) is preceded by the NP (which is assigned the `obj1` function). Here we use the operator < to implement *precedence*.

```
//node[@cat="pp" and
       node[@rel="obj1"]/number(@begin)
     < node[@rel="hd"]/number(@begin) ]
```

Note that we use in these examples the `number()` function to map the string value explicitly to a number. This is required in some implementations of XPath.

The operator = can be used to implement *direct precedence*. As another example, consider the problem of finding a prepositional phrase which follows a finite verb directly in a subordinate finite sentence. Initially, we arrive at the following query:

```
//node[@cat="ssub" and
       node[@rel="hd"]/number(@end)
     = node[@cat="pp"]/number(@begin)]
```

This does identify subordinate finite sentences in which the finite verb is directly followed by a PP. But note that the query also requires that this PP is a dependent of the same node. If we want to find a PP anywhere, then the query becomes:

```
//node[@cat="ssub" and
       node[@rel="hd"]/number(@end)
     = //node[@cat="pp"]/number(@begin)]
```

9.3.1.3 Pitfalls

The content and sub-structure of coindexed nodes (to represent secondary edges) is present in the XML structure only once. The index attribute is used to indicate equivalence of the nodes. This may have some unexpected effects. For instance, the following query will *not* match with the dependency structure given in Fig. 9.1.

```
//node[node[@rel="hd" and @lemma="stoppen"] and
       node[@rel="su" and @lemma="hij"]]
```

The reason is, that the subject of `stoppen` itself does not have a subject with `lemma=hij`. Instead, it does have a subject which is co-indexed with a node for which this requirement is true. In order to match this case also, the query should be complicated, for instance as follows:

```
//node[node[@rel="hd" and @lemma="stoppen"] and
       node[@rel="su" and
            ( @lemma="hij" or
              @index=//node[@lemma="hij"]/@index
            )]
```

The example illustrates that the use of co-indexing is not problematic for XPath, but it does complicate the queries in some cases. Some tools (for instance the Dact tool described in Sect. 9.3.3) provide the capacity to define macro substitutions in queries, which simplifies matters considerably.

9.3.2 Comparison with Lai and Bird 2004

In [6] a comparison of a number of existing query languages is presented, by focussing on seven example queries. Here we show that each of the seven queries can be formulated in XPath for the Lassy treebank. In order to do this, we first adapted the queries in a non-essential way. For one thing, some queries refer to English words which we mapped to Dutch words. Some other differences are that there is no (finite) VP in the Lassy treebank. The adapted queries with the implementation in XPath is now given as follows:

1. Find sentences that include the word zag.

   ```
   //node[@word="zag"]
   ```

2. Find sentences that do not include the word zag.

   ```
   not(//node[@word="zag"])
   ```

3. Find noun phrases whose rightmost child is a noun.

```
//node[@cat="np" and node[@pt="n"]/number(@end)=number(@end)]
```

4. Find root sentences that contain a verb immediately followed by a noun phrase that is immediately followed by a prepositional phrase.

```
//node[@cat="smain" and node[@pt="ww"]/number(@end)
    = node[@cat="np"
           and number(@end)
             = //node[@cat="pp"]/number(@begin)
          ]/number(@begin)]
```

5. Find the first common ancestor of sequences of a noun phrase followed by a prepositional phrase.

```
//node[.//node[@cat="np"]/number(@end) =
       .//node[@cat="pp"]/number(@begin) and
   not(node[.//node[@cat="np"]/number(@end) =
            .//node[@cat="pp"]/number(@begin)]) ]
```

6. Find a noun phrase which dominates a word `donker` (dark) that is dominated by an intermediate phrase that is a prepositional phrase.

```
//node[@cat="np" and
      .//node[@cat="pp" and .//node[@word="donker"]]]
```

7. Find a noun phrase dominated by a root sentence. Return the subtree dominated by that noun phrase only.

```
//node[@cat="smain"]/node[@cat="np"]
```

The ease with which the queries can be defined may be surprising to readers familiar with Lai and Bird [6]. In that paper, the authors conclude that XPath is not expressive enough for some queries. As an alternative, the special query language LPATH is introduced, which extends XPath in three ways:

- the additional axis *immediately following*
- the scope operator { . . . }
- the node alignment operators ^ and $

However, we note here that these extensions are unnecessary. As long as the surface order of nodes is explicitly encoded by XML attributes `begin` and `end`, as in the Lassy treebank, then the additional power is redundant. An LPATH query which requires that a node x immediately follows a node y can be encoded in XPath by requiring that the begin-attribute of x equals the end-attribute of y. The examples which motivate the introduction of the other two extensions likewise can be encoded in XPath by means of the begin- and end-attributes. For instance, the LPATH query

```
//node[@cat="smain"]{//node[@cat="np"]$}
```

where an SMAIN node is selected which contains a right-aligned NP can be defined in XPath as:

```
//node[@cat="smain" and number(@end) =
     .//node[@cat="np"]/number(@end)]
```

Based on these examples we conclude that there is no motivation for an ad-hoc special purpose extension of XPath, but that instead we can safely continue to use the XPath standard.

9.3.3 A Graphical User Interface for Lassy

Dact is a recent easy-to-use open-source tool, available for multiple platforms, to browse and search through Lassy treebanks. It provides graphical tree visualizations of the dependency structures of the treebank, full XPath search to select relevant dependency structures in a given corpus and to highlight the selected nodes of dependency structures, simple statistical operations to generate frequency lists for any attributes of selected nodes, and sentence-based outputs in several formats to display selected nodes e.g. by bracketing the selected nodes, or by a keyword-in-context presentation. Dact can be downloaded from http://rug-compling.github.com/dact/.

For the XML processing, Dact supports both the libxml2 (http://xmlsoft.org) and the Oracle Berkeley DB XML (http://www.oracle.com) libraries. In the latter case, database technology is used to preprocess the corpus for faster query evaluation. In addition, the use of XPath 2.0 is supported. Furthermore, Dact provides macro expansion in XPath queries.

The availability of XPath 2.0 is useful in order to specify quantified queries (argued for in the context of the Lassy treebanks in [1]). As an example, consider the query in which we want to identify a NP which contains a VC complement (infinite VP complement), in such a way that there is a noun which is preceded by the head of that NP, and which precedes the VC complement. In other words, in such a case there is an (extraposed) VC complement of a noun for which there is another noun which appears in between the noun and the VC complement. The query can be formulated as:

```
//node[@cat="np" and
      ( some $interm in //node[@pos="noun"]
   satisfies (   $interm/number(@begin)
               < node[@rel="vc"]/number(@begin) and
                 $interm/number(@end)
               > node[@rel="hd"]/number(@end)
             ))]
```

The availability of a macro facility is useful to build up more complicated queries in a transparent way. The following example illustrates this point. Macro's are defined using the format `name = string`. A macro is used by putting the name between `% %`. The following set of macro's defines the solution to the fifth problem posed in [6] in a more transparent manner. In order to define the minimal node which dominates a NP PP sequence, we first define the notion *dominates a NP PP sequence*, and then use it to state that the first common ancestor of a sequence of NP

PP is a node which is an ancestor of a NP PP sequence, but which does not contain a node which is an ancestor of a NP PP sequence.

```
b = number(@begin)
e = number(@end)
dominates_np_pp_seq =
  .//node[@cat="np"]/%e% = .//node[@cat="pp"]/%b%
q5 = //node[%dominates_np_pp_seq% and
            not(node[%dominates_np_pp_seq%])]
```

9.4 Using the Lassy Treebanks

9.4.1 Introduction

The availability of manually constructed treebanks for research and development in natural language processing is crucial, in particular for training statistical syntactic analysers or statistical components of syntactic analysers of a more hybrid nature. In addition such high quality treebanks are important for evaluation purposes for any kind of automatic syntactic analysers.

Syntactically annotated corpora of the size of Lassy Small are also very useful resources for corpus linguists. Note that the size of Lassy Small (one million words) is the same as the subset of the Corpus of Spoken Dutch (CGN) which has been syntactically annotated. Furthermore, the syntactic annotations of the CGN are also available in a format which is understood by Dact. This implies that it is now straightforward to perform corpus linguistic research both on spoken and written Dutch. Below, we provide a simple case study where we compare the frequency of WH-questions formed with `wie` (who) as opposed to `welk(e)` (which).

It is less obvious whether large quantity, lower quality treebanks are a useful resource. As one case in point, we note that a preliminary version of the Lassy Large treebank was used as *gold standard* training data to train a memory-based parser for Dutch [12]. In this article, we illustrate the expected quality of the automatic annotations, and we discuss an example study which illustrates the promise of large quantity, lower quality treebanks. In this section, we therefore focus on the use of the Lassy Large treebank.

9.4.2 Estimating the Frequency of Question Types

As an example of the use of Lassy Small, we report on a question of a researcher in psycholinguistics who focuses on the linguistic processing of WH-questions from a behavioral (e.g. self-paced reading studies) and neurological (event-related potentials) viewpoint. She studies the effect of information load: the difference between `wie` and `welk(e)` in for example:

Table 9.1 number of hits per query

	CGN		Lassy	
	#	%	#	%
wie	201	57.9	61	64.2
welk(e)	146	42.1	34	35.8

(2) Wie bakt het lekkerste brood?
Who bakes the nicest bread?

(3) Welke bakker bakt het lekkerste brood?
Which baker bakes the nicest bread?

To be sure that the results she finds are psycholinguistic or neurolinguistic in nature, she wants to be able to compare them to a frequency count in corpora.

Such questions can now be answered using the Lassy Small treebank or the CGN treebank by posing two simple queries. The following query finds WH-questions formed with `wie`:

```
//node[@cat="whq" and node[@rel="whd" and
      ( @lemma="wie" or .//node[@lemma="wie"] )]]
```

The number of hits of the queries are given in Table 9.1:

9.4.3 Estimation of the Quality of Lassy Large

In order to judge the quality of the Lassy Large corpus, we evaluate the automatic parser that was used to construct Lassy Large on the manually verified annotations of Lassy Small. The Lassy Small corpus is composed of a number of sub-corpora. Each sub-corpus is composed of a number of documents. In the experiment, Alpino (version of October 1, 2010) was applied to a single document, using the same options which have been used for the construction of the Lassy Large corpus. With these options, the parser delivers a single parse, which it believes is the best parse according to a variety of heuristics. These include the disambiguation model and various optimizations of the parser presented in [9, 16, 17]. Furthermore, a time-out is enforced in order that the parser cannot spend more than 190 s on a single sentence. If no result is obtained within this time, the parser is assumed to have returned an empty set of dependencies, and hence such cases have a very bad impact on accuracy.

In the presentation of the results, we aggregate over sub-corpora. The various `dpc-` sub-corpora are taken from the Dutch Parallel Corpus, and meta-information should be obtained from that corpus. The various `WR-` and `WS` corpora are inherited from D-COI. The `wiki-` subcorpus contains wikipedia articles, in many cases about topics related to Flanders.

Parsing results are listed in Table 9.2. Mean accuracy is given in terms of the f-score of named dependency relations. As can be observed from this table, parsing accuracies are fairly stable across the various sub-corpora. An outlier is the result of the parser on the WR-P-P-G sub-corpus (legal texts), both in terms of

Table 9.2 Parsing results (f-score of named dependencies) on the Lassy Small sub-corpora

Name	f-score	ms	#sent	Length	Name	f-score	ms	#sent	Length
dpc-bal-	92.77	1,668	620	14.2	dpc-bmm-	87.81	4,096	794	19.6
dpc-cam-	91.78	2,913	508	19.6	dpc-dns-	90.48	1,123	264	14.5
dpc-eli-	89.81	4,453	603	18.8	dpc-eup-	89.88	8,642	233	26.1
dpc-fsz-	85.74	4,492	574	19.1	dpc-gaz-	88.51	3,410	210	18.1
dpc-ibm-	90.13	4,753	419	20.2	dpc-ind-	91.14	4,010	1,650	20.6
dpc-kam-	89.82	4,671	52	25.6	dpc-kok-	88.00	2,546	101	18.3
dpc-med-	90.28	3,906	650	20.9	dpc-qty-	89.86	7,044	618	22.2
dpc-riz-	86.61	4,926	210	20.1	dpc-rou-	91.50	2,218	1,356	16.7
dpc-svb-	89.69	1,939	478	15.8	dpc-vhs-	90.83	1,819	461	14.4
dpc-vla-	90.57	2,545	1,915	16.8	wiki	88.85	1,940	7,341	13.4
WR-P-E-C	84.77	1,827	1,014	12.1	WR-P-E-E	82.61	3,599	90	20.1
WR-P-E-H	88.10	2,110	2,832	11.4	WR-P-E-I	87.78	4,051	9,785	20.4
WR-P-E-J	87.69	5,276	699	21.5	WR-P-P-B	92.07	318	275	7.3
WR-P-P-C	88.08	2,089	5,648	14.8	WR-P-P-E	89.14	3,759	306	19.0
WR-P-P-F	83.11	4,362	397	16.4	WR-P-P-G	80.32	10,410	279	23.2
WR-P-P-H	91.42	2,109	2,267	16.4	WR-P-P-I	90.43	3,369	5,789	20.0
WR-P-P-J	86.79	6,278	1,264	23.8	WR-P-P-K	89.37	3,715	351	19.9
WR-P-P-L	88.70	3,406	1,115	18.5	WS	90.40	1,596	14,032	14.7
Total	89.17	2,819	65,200	16.8					

accuracy and in terms of parsing times. We note that the parser performs best on the dpc-bal- subcorpus, a series of speeches by former prime-minister Balkenende.

9.4.4 The Distribution of zelf and zichzelf

As a further example of the use of parsed corpora to further linguistic insights, we consider a recent study [2] of the distribution of weak and strong reflexive objects in Dutch.

If a verb is used reflexively in Dutch, two forms of the reflexive pronoun are available. This is illustrated for the third person form in the examples below.

(4) Brouwers schaamt **zich/*zichzelf** voor zijn schrijverschap.
Brouwers shames *self1/self2* for his writing
Brouwers is ashamed of his writing

(5) Duitsland volgt ***zich/zichzelf** niet op als Europees kampioen.
Germany follows *self1/self2* not PART as European Champion
Germany does not succeed itself as European champion

(6) Wie **zich/zichzelf** niet juist introduceert, valt af.
Who self1/self2 not properly introduces, is out
Everyone who does not introduce himself properly, is out.

The choice between *zich* and *zichzelf* depends on the verb. Generally three groups of verbs are distinguished. Inherent reflexives never occur with a non-reflexive argument and occur only with *zich*(4). Non-reflexive verbs seldom, if ever occur with a reflexive argument. If they do, however, they can only take *zichzelf* as a reflexive argument (5). Accidental reflexives can be used with both *zich* and *zichzelf*, (6). Accidental reflexive verbs vary widely as to the frequency with which they occur with both arguments. [2] set out to explain this distribution.

The influential theory of [10] explains the distribution as the surface realization of two different ways of reflexive coding. An accidental reflexive that can be realized with both *zich* and *zichzelf* is actually ambiguous between an inherent reflexive and an accidental reflexive (which always is realized with *zichzelf*). An alternative approach is that of [3, 4, 11], who have claimed that the distribution of weak vs. strong reflexive object pronouns correlates with the proportion of events described by the verb that are self-directed vs. other-directed.

In the course of this investigation, a first interesting observation is, that many inherently reflexive verbs, which are claimed not to occur with *zichzelf*, actually often do combine with this pronoun. Two typical examples are:

(7) Nederland moet stoppen zichzelf op de borst te slaan
Netherlands must stop self2 on the chest to beat
The Netherlands must stop beating itself on the chest

(8) Hunze wil zichzelf niet al te zeer op de borst kloppen
Hunze want self2 not all too much on the chest knock
Hunze doesn't want to knock itself on the chest too much

With regards to the main hypothesis of their study, Bouma and Spenader [2] use linear regression to determine the correlation between reflexive use of a (non-inherently reflexive) verb and the relative preference for a weak or strong reflexive pronoun. Frequency counts are collected from the parsed TwNC corpus (almost 500 million words). They limit the analysis to verbs that occur at least 10 times with a reflexive meaning and at least 50 times in total, distinguishing uses by subcategorization frames. The statistical analysis shows a significant correlation, which accounts for 30 % of the variance of the ratio of nonreflexive over reflexive uses.

9.5 Validation

The Lassy Small and Lassy Large treebanks have been validated by a project-external body, the Center for Sprogteknologi, University of Copenhagen. The validation report gives a detailed account of the validation of the linguistic annotations of syntax, PoS and lemma in the Lassy treebanks. The validation comprises extraction of validation samples, manual checks of the content, and computation of named dependency accuracy figures of the syntax validation results.

The content validation falls in two parts: validation of the linguistic annotations (PoS-tagging, lemmatization) and the validation of the syntactic annotations. The validation of the syntactic annotation was carried out on 250 sentences from Lassy Large and 500 sentences from Lassy Small, all randomly selected. The validation of the lemma and PoS-tag annotations was carried out on the same sample from Lassy Small as for syntax, i.e. 500 sentences.

Formal validation i.e. the checking of formal information such as file structure, size of files and directories, names of files etc. is not included in this validation task but no problems were encountered in accessing the data and understanding the structure. For the syntax, the validators computed a sentence based accuracy (number of sentences without errors divided by the total number of sentences). For Lassy Large, the validators found that the syntactic analysis was correct for a proportion of 78.4 % of the sentences. For Lassy Small, the proportion of correct syntactic analyses was 97.8 %. Out of the 500 validated sentences with a total of 8,494 words, the validators found 31 words with a wrong lemma (the accuracy of the lemma annotation therefore is 99.6 %. For this same set of sentences, validators found 116 words with wrong part-of-speech tag (accuracy 98.63 %).

In conclusion, the validation states that the Lassy corpora comprise a well elaborated resource of high quality. Lassy Small, the manually verified corpus, has really fine results for both syntax, part-of-speech and lemma, and the overall impression is very good. Lassy Large also has fine results for the syntax. The overall impression of the Lassy Large annotations is that the parser succeeds in building up acceptable trees for most of the sentences. Often the errors are merely a question of the correct labeling of the nodes.

9.6 Conclusion

In this article we have introduced the Lassy treebanks, and we illustrated the lemma, part-of-speech and dependency annotations. The quality of the annotations has been confirmed by an external validation. We provided evidence that the use of the standard XPath language suffices for the identification of relevant nodes in the treebanks, countering some evidence to the contrary by Lai and Bird [6] and Bouma [1]. We illustrated the potential usefulness of Lassy Small by estimating the frequency of question types in the context of a psycho-linguistic study. We furthermore illustrated the use of the Lassy Large treebank in a study of the distribution of the two Dutch reflexive pronouns `zich` and `zichzelf`.

Appendices

A. List of Category Labels

Label	Explanation	Label	Explanation
AP	Adjectival phrase	ADVP	Adverbial phrase
AHI	*aan het*-infinitive VP	CONJ	Conjunction
CP	Subordinate sentence	DETP	Determiner phrase
DU	Discourse unit	INF	Bare infinitive VP
NP	Noun phrase	OTI	*om te*-infinitive VP
PPART	Past participle VP	PP	Prepositional phrase
PPRES	Present participle VP	REL	Relative clause
SMAIN	Root sentence	SSUB	Subordinate finite VP
SVAN	*van* finite VP	SV1	Verb-first phrase (yes/no question, imperative)
TI	*te*-infinitive VP	WHREL	Free relative clause
WHSUB	Embedded question	WHQ	WH-question

B. List of Dependency Labels

Label	Explanation	Label	Explanation
APP	Apposition		
BODY	Body	CMP	Complementizer
CNJ	Conjunct	CRD	Coordinator
DET	Determiner	DLINK	Discourse-link
DP	Discourse-part	HD	Head
HDF	Post-position	LD	Locative/directional complement
ME	Measure complement	MOD	Modifier
MWP	Part of multi-word-unit	NUCL	Nucleus
OBCOMP	Complement of comparison	OBJ1	Direct object
OBJ2	Secondary object	PC	Prepositional complement
POBJ1	Temporary direct object	PREDC	Predicative complement
PREDM	Predicative modifier	RHD	Head of relative clause
SAT	Satellite	SE	Reflexive object
SU	Subject	SUP	Temporary subject
SVP	Particle	TAG	Tag
VC	Verbal complement	WHD	Head of WH-question

References

1. Bouma, G.: Starting a Sentence in Dutch. Ph.D. thesis, University of Groningen (2008)
2. Bouma, G., Spenader, J.: The distribution of weak and strong object reflexives in Dutch. In: van Eynde, F., Frank, A., Smedt, K.D., van Noord, G. (eds.) Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7), no. 12 in LOT Occasional Series, pp. 103–114. Netherlands Graduate School of Linguistics, Utrecht, The Netherlands (2009)
3. Haspelmath, M.: A frequentist explanation of some universals of reflexive marking (2004). Draft of a paper presented at the Workshop on Reciprocals and Reflexives, Berlin
4. Hendriks, P., Spenader, J., Smits, E.J.: Frequency-based constraints on reflexive forms in Dutch. In: Proceedings of the 5th International Workshop on Constraints and Language Processing, pp. 33–47. Roskilde, Denmark (2008). http://www.ruc.dk/dat_en/research/reports
5. Hoekstra, H., Moortgat, M., Schouppe, M., Schuurman, I., van der Wouden, T.: CGN Syntactische Annotatie (2004). http://www.tst-centrale.org/images/stories/producten/documentatie/cgn_website/doc_Dutch/topics/annot/syntax/syn_prot.pdf
6. Lai, C., Bird, S.: Querying and updating treebanks: a critical survey and requirements analysis. In: In Proceedings of the Australasian Language Technology Workshop, pp. 139–146. Sydney, Australia (2004)
7. Oostdijk, N., Reynaert, M., Monachesi, P., van Noord, G., Ordelman, R., Schuurman, I., Vandeghinste, V.: From D-Coi to SoNaR. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)
8. Pajas, P., Štěpánek, J.: Recent advances in a feature-rich framework for treebank annotation. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 673–680. Coling 2008 Organizing Committee, Manchester, UK (2008). http://www.aclweb.org/anthology/C08-1085
9. Prins, R., van Noord, G.: Reinforcing parser preferences through tagging. Traitement Automatique des Langues **44**(3), 121–139 (2003)
10. Reinhart, T., Reuland, E.: Reflexivity. Linguist. Inq. **24**, 656–720 (1993)
11. Smits, E.J., Hendriks, P., Spenader, J.: Using very large parsed corpora and judgement data to classify verb reflexivity. In: Branco, A. (ed.) Anaphora: Analysis, Algorithms and Applications, pp. 77–93. Springer, Berlin (2007)
12. van den Bosch, A., Busser, B., Canisius, S., Daelemans, W.: An efficient memory-based morphosyntactic tagger and parser for Dutch. In: Dirix,P., Schuurman, I., Vandeghinste, V., van Eynde, F. (eds.) Computational Linguistics in the Netherlands 2006. Selected Papers from The Seventeenth CLIN meeting, LOT Occassional Series, pp. 99–114. LOT Netherlands Graduate School of Linguistics, Utrecht, The Netherlands. Leuven, Belgium (2007)
13. van Eerten, L.: Over het Corpus Gesproken Nederlands. Nederlandse Taalkunde **12**(3), 194–215 (1997)
14. Van Eynde, F.: Part Of Speech Tagging En Lemmatisering Van Het D-Coi Corpus (2005). http://www.let.rug.nl/~vannoord/Lassy/POS_manual.pdf
15. van Noord, G.: **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In: TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles, Leuven, pp. 20–42 (2006)
16. van Noord, G.: Learning efficient parsing. In: EACL 2009, The 12th Conference of the European Chapter of the Association for Computational Linguistics, Athens, Greece, pp. 817–825 (2009)

17. van Noord, G., Malouf, R.: Wide coverage parsing with stochastic attribute value grammars (2005). Draft available from the authors. A preliminary version of this paper was published in the Proceedings of the IJCNLP workshop Beyond Shallow Analyses, Hainan, China (2004)
18. van Noord, G., Schuurman, I., Bouma, G.: Lassy syntactische annotatie, revision 19455 (2011). http://www.let.rug.nl/vannoord/Lassy/sa-man_lassy.pdf

Chapter 10
Cornetto: A Combinatorial Lexical Semantic Database for Dutch

Piek Vossen, Isa Maks, Roxane Segers, Hennie van der Vliet, Marie-Francine Moens, Katja Hofmann, Erik Tjong Kim Sang, and Maarten de Rijke

10.1 Introduction

One of the goals of the STEVIN programme is the realisation of a digital infrastructure that will enforce the position of the Dutch language in the modern information and communication technology. A semantic database for Dutch is a crucial component for this infrastructure for three reasons: (1) it enables the development of semantic web applications on top of knowledge and information expressed in Dutch, (2) it provides people with access to information systems and services through their native Dutch language and (3) it will connect the Dutch language to the semantic

P. Vossen (✉) · I. Maks · H. van der Vliet
Faculty of Arts, VU University of Amsterdam, De Boelelaan 1105, 1081 HV, Amsterdam, The Netherlands
e-mail: piek.vossen@vu.nl; e.maks@vu.nl; h.d.vander.vliet@vu.nl

R. Segers
Faculty of Sciences, VU University of Amsterdam, De Boelelaan 1105, 1081 HV, Amsterdam, The Netherlands
e-mail: r.h.segers@vu.nl

M.-F. Moens
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium
e-mail: Sien.Moens@cs.kuleuven.be

M. de Rijke · K. Hofmann
ISLA, University of Amsterdam, Science Park 904, 1098 XH, Amsterdam, The Netherlands
e-mail: derijke@uva.nl; k.hofmann@uva.nl

E. Tjong Kim Sang
Faculteit der Letteren, University of Groningen, Oude Kijk in 't Jatstraat 26, 9712 EK, Groningen, The Netherlands
e-mail: erik@xs4all.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_10,

processing of English knowledge and information. A semantic database makes it possible to go from words to concepts and consequently, to develop technologies that access and use knowledge rather than textual representations.

At the start of STEVIN, there were two separate semantic databases for contemporary Dutch: the Referentiebestand Nederlands (RBN, [20, 21]) and the Dutch wordnet (DWN, [32, 33]). These databases contain partially overlapping and partially complementary information. More importantly, they represent different perspectives on the semantics of words: RBN follows a word-to-meaning perspective that differentiates the meanings of words in terms of their combinatoric behavior, while DWN follows a meaning-to-word perspective that defines words with the same meaning as a single concept through semantic relations between these concepts. The goal of the Cornetto project was to combine these two database into a single unique semantic resource with both the rich semantic relations taken from DWN and the typical combinatoric lexical constraints, as reflected in multiword expressions, idioms, collocations and frames taken from RBN. However, Cornetto nevertheless maintains both perspectives in the same database by representing the data as two separate but linked collections. Likewise, Cornetto can be used to view word meanings in both ways, which will eventually lead to a better and more consistent definition of the similarities and differences of the semantics and usage of words. Since the meaning-to-word view is structured according to and linked to Princeton Wordnet (PWN) [14], the semantics of the database is open to technologies developed for English. This enables transferring state-of-the-art language technologies from English to Dutch, such as semantic similarity measurement, query expansion and automatic word-sense-disambiguation.

The Cornetto database[1] was built by automatically aligning the word meanings of both databases on the basis of the overlapping information and next revising these mappings using an editor that was developed during the project. The final database contains over 92K lemmas (70K nouns, 9K verbs, 12K adjectives and 73 adverbs) corresponding to 118K word meanings. Through the alignment with PWN, ontological and domain labels were imported. In addition to the database, there is a toolkit for the acquisition of new concepts and relations, and the tuning and extraction of a domain specific sub-lexicon from a compiled corpus. Such a sub-lexicon is extracted for the domain of financial law. The Cornetto database is owned by the Dutch Language Union (Nederlandse Taalunie, NTU) and is free of charge for research.[2]

The remainder of this article is organised as follows, in Sect. 10.2 we describe work related to combining lexical resources. In Sect. 10.3 we specify the design of the database and in Sect. 10.4 we elaborate on the techniques that have been used to align RBN and DWN. In Sect. 10.5 we explain the manual editing phase

[1] http://www2.let.vu.nl/oz/cltl/cornetto/index.html

[2] Licenses can be obtained from: http://www.tst-centrale.org/nl/producten/lexica/cornetto/7-56. An external evaluation was carried out by Polderland [10]. For commercial usage, a fee must be paid to the NTU for background data that is included.

and in Sect. 10.6 we present the qualitative and quantitative results. Additionally, in Sect. 10.7 we present two acquisition toolkits that have been developed. In Sect. 10.8 we present an overview of the current use of Cornetto and finally in Sect. 10.9 we conclude with observations and lessons learned.

10.2 Related Work

In order to optimise the reusability of lexical knowledge in various resources, combining these resources becomes crucial. Many attempts involve combining lexical resources with morpho-syntactic information, e.g. [5, 12, 23, 25]. This is however a different task than matching semantic resources because it involves a finite set of specifications of limited morpho-syntactic properties instead of a large set of concepts. Once these morpho-syntactic specifications are aligned, matching lexical entries is rather trivial. One of the first approaches to semantically align lexicons was proposed in the Acquilex project [1, 6], using so-called t-links across monolingual lexical knowledge bases. These knowledge bases contain very rich and detailed typed-feature structure representations, which make matching across the specifications a form of unification. The actual links were only generated for small specialised lexicons.

One of the earliest attempts of large scale alignment was done in the EuroWordNet project [32]. In this case, wordnets in other languages are aligned with the PWN using the semantic relations in each and translations from a bilingual dictionary [13] and [2]. Although this type of alignment is cross-lingual, we used similar techniques for Cornetto but in a monolingual context and using less semantic relations. Other work on large scale alignment for monolingual resources is described in [31] and [30] for ontology alignment. This is a relatively easy task due to the rich hierarchical structure and the lack of polysemy. More complex is the work of [19], who try to align FrameNet [3] and PWN. This type of alignment comes closer to the problem addressed in Cornetto, since both are large monolingual resources with detailed descriptions of different meanings (high polysemy) and having different semantic structures.

The Cornetto project is also related to more recent work on the development of the ISO standard for lexical resources (LMF[3], ISO-24613:2008) and Wordnet-LMF [29]. Especially, Wordnet-LMF, an extension of LMF to include wordnet data, benefited from the work done in Cornetto. In Cornetto, separate collections and representations are maintained for the RBN part and the DWN part. The RBN part can be converted to an LMF representation for word meanings, while the DWN part can be structured as Wordnet-LMF, combining the benefits of both.

[3] www.lexicalmarkupframework.org/

10.3 The Design of the Database

Both DWN and RBN are semantic lexical resources. RBN uses a traditional structure of form-meaning pairs, so-called Lexical Units. Lexical Units (LUs) are word senses in the lexical semantic tradition. They contain the linguistic knowledge that is needed to properly use the word in a specific meaning in a language. Since RBN follows a word-to-meaning view, the semantic and combinatoric information for each meaning typically clarify the differences across the meanings. RBN likewise focusses on the polysemy of words and typically follows an approach to represent condensed and generalised meanings from which more specific ones can be derived.

On the other hand, DWN is organised around the notion of *synsets*. Synsets are sets of synonyms that represent a single concept as defined by [14], e.g. *box* and *luidspreker* in Dutch are synonyms for *loud speaker*. Synsets are conceptual units based the lexicalisations in a language.[4] In Wordnet, concepts are defined in a graph by lexical semantic relations, such as hypernyms (broader term), hyponyms (narrower term), role relations. Typically in Wordnet, information is provided for the synset as a whole and not for the individual synonyms, thus presenting a meaning-to-word view on a lexical database and focussing on the similarities of word meanings. For example, word meanings that are synonyms have a single gloss or definition in Wordnet but have separate definitions in RBN as different lexical units. From a Wordnet point of view, the definitions of LUs from the same synset should be semantically equivalent and the LUs of a single word should belong to different synsets. From a RBN point of view, the LUs of a single word typically differ in terms of connotation, pragmatics, syntax *and* semantics but synonymous words of the same synset can be differentiated along connotation, pragmatics and syntax but not semantics.

Outside the lexicon, an ontology provides a third layer of meaning. In Cornetto, SUMO [24] has been used as the ontological framework. SUMO provides good coverage, is publicly available, and all synsets in PWN are mapped to it. Through the equivalence relations from DWN to PWN, mappings to SUMO can be imported automatically.[5] The concepts in an ontology are referred to as Terms. Terms represent types that can be combined in a knowledge representation language to form axioms. In principle, Terms are defined independently of language but according to principles of logic. In Cornetto, the ontology represents an independent anchoring of the pure relational meaning in Wordnet. The ontology is a formal framework that can be used to constrain and validate the implicit semantic statements of the lexical semantic structures, both for LUs and synsets. Further, the semantic anchoring to the ontology contributes to the development of semantic web applications for which language-specific lexicalisations of ontological types are useful.

[4] As such, Wordnets for different languages show a certain level of idiosyncrasy.

[5] For more information about SUMO please refer to http://www.ontologyportal.org/

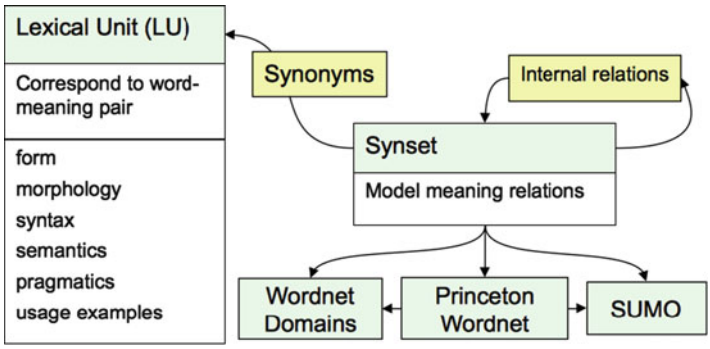


Fig. 10.1 Data collections in the Cornetto Database

A fourth layer is represented by Wordnet Domains [22]. Domains represent clusters of concepts that are related by a shared area of interest, such as *sport*, *education* or *politics*. Whereas different instruments can be subclasses of the same ontological Term (e.g. *tank* and *ambulance* are both of the type *Vehicle*), they may belong to different Domains (e.g. *military* and *medical*).

The Cornetto database (CDB) thus consists of 4 layers of information represented in two collections:

1. Collection of Lexical Units (LU), mainly derived from the RBN
2. Collection of Synsets, derived from DWN with mappings to PWN
3. Mappings to Terms and axioms in SUMO
4. Mappings to Domains in Wordnet Domains

Figure 10.1 shows an overview of the different data structures and their relations. There may be LUs that do not occur in synsets but there are no synonyms in synsets that are not LUs. The synsets are organised by means of internal relations such as hypernyms, while the LUs provide rich information on morphology, syntax and pragmatics. The synsets also point to external sources: the Princeton Wordnet (PWN), Wordnet domains (DM) and the SUMO ontology. The Cornetto database is implemented in the Dictionary Editor and Browser (DEB II) platform [18], while the raw XML files are distributed by the TST centrale. The XML Schema file for the data can be downloaded from the Cornetto website.

Figure 10.2 provides a simplified overview of the interplay between the different data structures. Here, four meanings of *band* are defined according to their semantic relations in DWN, RBN, SUMO and Wordnet Domains. Black arrows represent hypernym relations while the dashed arrows represent other semantic relations such as a Mero-Member between ‘music group’ and ‘musician’. Note that the hypernym of each synset for *band* is similar to SUMO terms, e.g. *middel* (device) and Device. However, the SUMO terms are fully axiomatised externally, while the implications of the hypernym relation remain implicit.

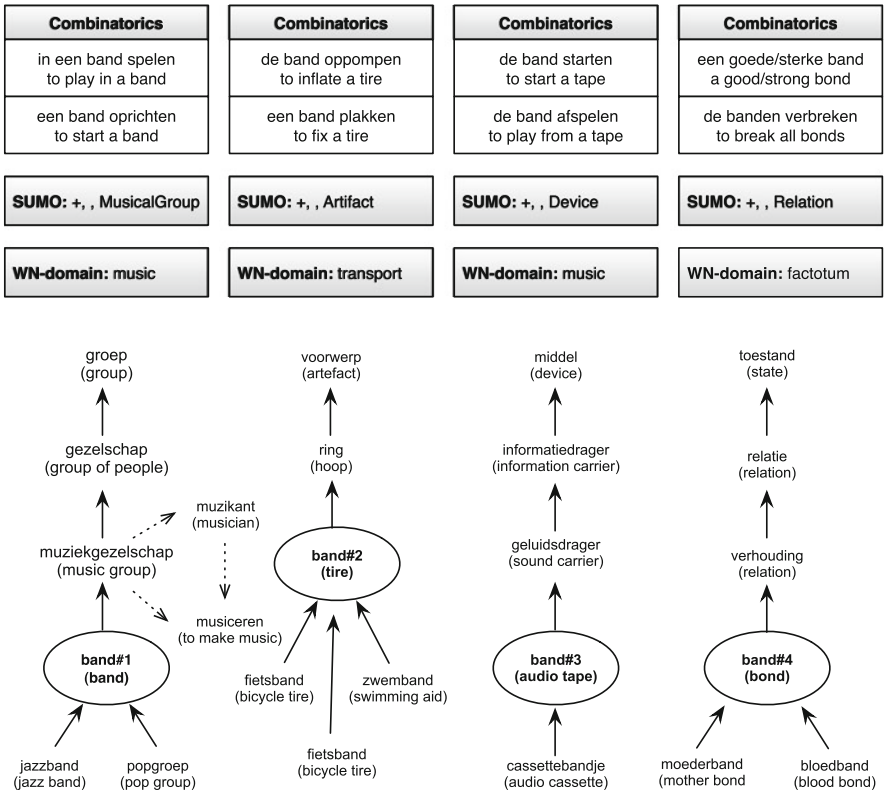


Fig. 10.2 Simplified example of the combinatorics and semantic relations for the word *band*

In the next sections, we describe the data collections for the synsets, the lexical units and the mappings to SUMO terms in more detail.

10.3.1 Lexical Units

The data structure for the LUs is implemented as a list; every LU element has an unique identifier or c_lu_id. The database for LUs contains structures to represent the form, syntactic and morphological information, semantics, pragmatics, and usage examples. An example of the XML structure for the first sense of the noun *band* (tire) is shown in Fig. 10.3. The xml of this LU contains basic morpho-syntactic information (lines 3–8), some semantics (lines 11–15) and additional examples on the combinatorial behaviour of the word such as the lexical collocation *de band oppompen* (to inflate a tire) at line 41, and an idiomatic usage: *uit de band springen* (excessive behavior) at line 20.

```
<cdb_lu c_seq_nr="1" type="swu" is_complete="true" c_lu_id="r_n-5873">
  <form form-cat="noun" form-spelling="band"/>
  <morphology_noun>
    <morpho-type>simpmorph</morpho-type>
    <morpho-plurforms><morpho-plurform>banden</morpho-plurform>
    </morpho-plurforms>
  </morphology_noun>
  <syntax_noun><sy-gender>m</sy-gender> <sy-article>de</sy-article>
  </syntax_noun>
  <semantics_noun>
    <sem-reference>common</sem-reference>
    <sem-countability>count</sem-countability>
    <sem-type>artefact</sem-type>
    <sem-subclass>vervoermiddel (deel v.)</sem-subclass>
    <sem-resume>om een wiel</sem-resume>
  </semantics_noun>
  <examples>
  <example r_ex_id="37490">
    <form_example>
      <canonicalform>uit de band springen</canonicalform>
      <category>vp</category>
    </form_example>
    <syntax_example>
      <sy-type>fixed</sy-type>
      <sy-subtype>idiom</sy-subtype>
      <sy-combi>
        <sy-combipair>
          <sy-combiword>uit</sy-combiword> <sy-combicat>prep</sy-combicat>
        </sy-combipair>
        <sy-combipair>
          <sy-combiword>springen</sy-combiword> <sy-combicat>verb</sy-combicat>
        </sy-combipair>
      </sy-combi>
    </syntax_example>
    <semantics_example>
      <sem-meaningdescription>zich laten gaan</sem-meaningdescription>
    </semantics_example>
  </example>
  <example r_ex_id="37491">
    <form_example>
      <canonicalform>de band oppompen</canonicalform>
      <category>vp</category>
    </form_example>
    <syntax_example>
      <sy-type>fixed</sy-type>
      <sy-subtype>lexcol</sy-subtype>
      <sy-combi>
        <sy-combipair>
          <sy-combiword>oppompen</sy-combiword> <sy-combicat>verb</sy-combicat>
        </sy-combipair>
      </sy-combi>
      </syntax_example>
    <semantics_example>
      <sem-meaningdescription>met een pomp lucht blazen in een rubber band
        zodat hij harder wordt</sem-meaningdescription>
      <sem-lc-collocator>causeupgra</sem-lc-collocator>
    </semantics_example>
  </example>
</examples>
</cdb_lu>
```

Fig. 10.3 Shortened example of the XML structure for the lexical unit *band*

For nouns, the morpho-syntactic information is relatively simple. Figure 10.4 shows the rich information provided for verbs, illustrated by the LU *oppompen* (to inflate). The syntax field (lines 12–16) specifies the transitivity, valency and complementation of this verb. The semantics field provides information about the caseframe (lines 20–28); *oppompen* is an action verb with a selection restriction on the agent (animate agent) and no further restrictions on the theme. Finally, both a canonical (line 37) and a textual example (line 38) are given with typical fillers for the theme of this verb: 'tube', 'tire' and 'ball'. For a further description of the structure and contents, we refer to the Cornetto deliverable [11].

```
<cdb_lu c_seq_nr="1" type="swu" is_complete="true" c_lu_id="r_v-5716">
    <form form-cat="verb" form-spelling="oppompen"/>
    <morphology_verb>
        <morpho-type>phrasal </morpho-type>
        <morpho-structure >[op]pompen</morpho-structure >
        <flex-conjugation><flex-conjugationtype>regular </flex-conjugationtype>
          </flex-conjugation>
        <flex-mode>inf </flex-mode> <flex-tense>ntense </flex-tense>
        <flex-number>nnumber</flex-number><flex-person>nperson </flex-person>
    </morphology_verb>
    <syntax_verb>
        <sy-trans>tran </sy-trans><sy-separ>sch </sy-separ>
        <sy-class>main</sy-class><sy-peraux>h</sy-peraux>
        <sy-valency>di </sy-valency> <sy-reflexiv >nrefl </sy-reflexiv >
        <sy-subject>pers </sy-subject>
        <sy-complementation><sy-comp>np </sy-comp></sy-complementation>
    </syntax_verb>
    <semantics_verb>
        <sem-type>action </sem-type>
        <sem-caseframe>
            <caseframe>action2 </caseframe>
            <args>
                <arg><caserole>agent </caserole><selrestrole>agentanimate</selrestrole>
                  <synset_list/> </arg>
                <arg><caserole>theme </caserole> <selrestrole>themenselres </selrestrole>
                  <synset_list/></arg>
            </args>
        </sem-caseframe>
        <sem-resume>vol lucht blazen </sem-resume>
    </semantics_verb>
      <pragmatics>
    <prag-domain general="true" subjectfield="tech"/>
    </pragmatics>
      <examples>
          <example r_ex_id="13374">
          <form_example>
            <canonicalform>een fietsband/tube/bal oppompen</canonicalform>
            <textualform>hij heeft zijn fietsbanden nog eens stevig opgepompt en
                                    zijn ketting goed gesmeerd</textualform>
            <category>vp</category>
            <text-category>s</text-category>
          </form_example>
          <syntax_example>
            <sy-type>free </sy-type>
              <sy-combi>
              <sy-combipair>
                <sy-combiword>fietsband </sy-combiword> <sy-combicat>noun</sy-combicat>
              </sy-combipair>
              <sy-combipair>
                <sy-combiword>tube </sy-combiword> <sy-combicat>noun</sy-combicat>
              </sy-combipair>
            </sy-combi>
            </syntax_example>
            </example>
    </examples>
</cdb_lu>
```

Fig. 10.4 Shortened example of verbal lexical unit for *oppompen* (to inflate)

10.3.2 Synsets

Synsets are identified by an unique identifier or c_synset_id, which is used to reference synsets. An additional attribute, d_synset_id, links synsets to their source concepts in DWN in order to make the lookup for the alignment process more efficient. Each synset contains one or more synonyms; each of these synonym entries consists of a pointer to a LU (c_lu_id).

Figure 10.5 illustrates the structure in more detail for the synset *band*. It has *luchtband* (tire filled with air) as a synonym (lines 2–5). Further, the example shows that *band* has several semantic relations to other concepts such as a hypernym relation to *ring* (line 20) and to various instruments that apply to *tires*, such as

```
<cdb_synset c_sy_id="d_n-38252" posSpecific="NOUN_MASCULINE" d_synset_id="d_n-38252" comment="">
  <synonyms>
    <synonym status="rbn-1-dwn-1" c_cid_id="27346" c_lu_id-previewtext="luchtband:1"
      c_lu_id="r_n-22643"/>
    <synonym status="" c_cid_id="" c_lu_id-previewtext="band:1" c_lu_id="r_n-5873"/>
  </synonyms>
  <base_concept>false</base_concept>
  <definition>met lucht gevulde band voor voertuigen;om een wiel;</definition>
  <wn_internal_relations>
    <relation factive="" reversed="false" relation_name="CO_PATIENT_INSTRUMENT"
      target-previewtext="bandelichter:1, bandafnemer:1, bandwipper:1, bandenlichter:1"
      negative="false" coordinative="false" disjunctive="false" target="d_n-21407">
        <author name="piek" score="0.0" status="YES" date="19990301" source_id="d_n-38252"/>
    </relation>
    <relation factive="" reversed="false" relation_name="CO_PATIENT_INSTRUMENT"
      target-previewtext="bandrem:2" negative="false"
      coordinative="false" disjunctive="false" target="d_n-10174">
        <author name="Piek" score="0.0" status="" date="19961217" source_id="d_n-38252"/>
    </relation>
    <relation factive="" reversed="false" relation_name="HAS_HYPERONYM"
      target-previewtext="ring:2, ringetje:1"
      negative="false" coordinative="false" disjunctive="false" target="d_n-41726">
        <author name="Paul" score="0.0" status="" date="19961206" source_id="d_n-38252"/>
    </relation>
  </wn_internal_relations>
  <wn_equivalence_relations>
    <relation target20-target20Previewtext="tire:1, tyre:2" relation_name="EQ_SYNONYM"
      target15="ENG15-03192201-n" version="pwn-1.5" target30="ENG30-04440749-n"
      target20="ENG20-04269070-n">
        <author name="Laura" score="10523.0" status="YES" date="19980903" source_id=""/>
    </relation>
  </wn_equivalence_relations>
  <wn_domains>
    <dom_relation name="roxane" status="true" term="transport"/>
  </wn_domains>
  <sumo_relations>
    <ont_relation name="dwn10_pwn15_pwn20_mapping" status="false"
      relation_name="+" arg1="" arg2="Artifact"/>
  </sumo_relations>
</cdb_synset>
```

Fig. 10.5 Example of xml structure for the synset for *band* in its first sense

bandenlichter (tire lever) at line 10, and *bandrem* (tire brake) at line 15.[6] It also shows an EQ_SYNONYM relation to the English synset for *tire* at line 27, a relation to the domain *transport* at line 34 and a subclass relation (+) to the SUMO class *Artifact* at line 38.

10.3.3 SUMO Ontology Mappings

The SUMO ontology mappings provide the conceptual anchoring of the synsets and the lexical units. The mappings to Terms in SUMO have been imported from the equivalence relations of the synsets to PrincetonWordNet (PWN). Four basic relations are used in Princeton Wordnet and Cornetto:

=	The synset is equivalent to the SUMO concept
+	The synset is subsumed by the SUMO concept
@	The synset is an instance of the SUMO concept
[	The SUMO concept is subsumed by the synset

[6]For an overview of all semantic relations used in Cornetto, we refer to Cornetto deliverable D16.

The mappings from PWN to SUMO consist of two placeholders: one for the four relations (=, +, @, []) and one for the SUMO term. In Cornetto, we extended this representation with a third placeholder to define more complex mappings from synsets to the SUMO ontology. For this, the above relations have been extended with all relations defined in SUMO (version April 2006). The relation name and two arguments represent a so-called triple.[7] The arguments of the triples follow the syntax of the relation names in SUMO: the first slot is reserved for the relation, the second slot for a variable and the third slot contains either a SUMO term or an additional variable. The variables are expressed as integers, where the integer 0 is reserved to co-index with the referent of the synset that is being defined.

For example, the following expressions are possible in the Cornetto database:

1. Equality *cirkel* (circle): (=, 0, Circle)
2. Subsumption *band* (tire): (+, 0, Artifact)
3. Related *bot* (bone) : (part, 0, Skeleton)
4. Axiomatised *theewater* (tea water): ((instance, 0, Water) (instance, 1, Making) (instance, 2, Tea) (resource, 0, 1) (result, 2,1))

Relations directly imported from Princeton Wordnet will have the structure of 1 and 2. The triples in 3 and 4 are used to specify a complex mapping relation to the SUMO ontology, in case the basic mapping relations are not sufficient. This is especially the case for so-called non-rigid concepts [16], e.g. *theewater* (water used for making tea) is not a type of water but water used for some purpose. The triples given in 4 likewise indicate that the synset refers to an instance of Water rather than a subclass and that this instance is involved in the process of making Tea as a resource.[8]

10.4 Building the Database

The semantic units of the Cornetto database, whether LUs or synsets, are based on the word meaning distinctions that are made in RBN and DWN. The database is created by aligning these units while maintaining separate collections. The smallest semantic unit is used for making the alignment, which is the LU. The overall procedure for building the database consisted of (1) an automatic alignment to create mappings for the LUs from RBN and DWN and to generate the initial Cornetto collections and (2) a manual revision of the mappings. This procedure is illustrated in Fig. 10.6 for the word *koffie* (coffee). We see that it originally had four meanings in DWN and two in RBN. The two RBN meanings match with meanings 2 and 3

[7] Note that these triples should not be mistaken with RDF triples: the Cornetto ontology triples have no URIs.

[8] For further details on the SUMO mappings in Cornetto, see the deliverable. [11]

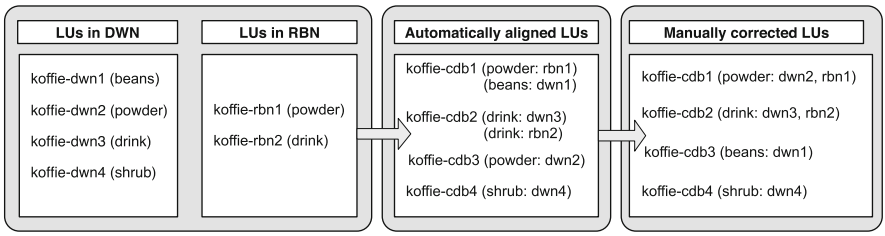


Fig. 10.6 The alignment procedure for the word *koffie* (coffee)

in DWN but the automatic procedure fails to match the second meaning (powder) of DWN and wrongly matches the first DWN meaning (beans) to the first meaning of RBN (powder). In the initial Cornetto database, we thus get four meanings for *koffie* but not all are correct. The manual revision then aligns the second DWN meaning (powder) with the first RBN meaning and creates a new LU for the first DWN meaning (beans).

The automatic alignment program initially created scored mappings across all LUs. The mappings are based on a number of heuristics taking into account: (1) the number of meanings, (2) overlapping definition words and synonyms, and (3) mappings of domains. For creating the merged database, the highest mapping relations are considered above a threshold that was empirically established from samples by eight native speakers. Precision scores varied from 54 to 97 % depending on the heuristics (more details can be found in [7]). The program created a minimal set of LUs and synsets as follows:

1. If there is a best scoring mapping between an LU in RBN and a synonym in DWN, create a single unique LU which becomes a synonym of a synset. The LU receives the ID from RBN and the synset receives the ID from DWN;
2. For all remaining mappings: do not create LUs and/or synsets in Cornetto but store additional mappings that can be accessed as weighted alternatives;
3. If there is no mapping for a LU in RBN to a synonym in DWN, create a unique LU in Cornetto with the RBN LU ID and do not create a synset for the LU in Cornetto;
4. If there is no mapping for a synonym in DWN to an LU in RBN, create (1) a synset in Cornetto with the DWN synset ID and (2) create a Cornetto LU with the DWN LU ID.

As a result, all LUs from RBN were thus copied to the Cornetto LU repository and all synsets from DWN were copied to the Cornetto synset repository. If an LU was mapped to an LU from DWN, this LU became a synonym in the DWN synset, replacing the original DWN LU. DWN LUs that could not be mapped to RBN LUs were added to the LU repository. Table 10.1 shows the degree of matching across the original resources RBN and DWN obtained through the automatic alignment. About 38 % of the LUs are matched. Almost 60 % consists of LUs from DWN not matched with RBN: mostly words not occurring in RBN. Similarly, 3,223 LUs from

Table 10.1 Number of matching and non-matching lexical units

Matches	Absolute	Relative (%)
DWN and RBN matches	35,289	37.74
LUs only in DWN	54,983	58.81
LUs only in RBN	3,223	3.45
Total	93,495	–

RBN could not be matched with a LU in DWN. For these, we did not create a new synset. The reason for this is that they often can be added manually as a synonym to an existing synset.[9]

10.5 Editing the Cornetto Database

The core Cornetto database was manually revised and checked, using an editing protocol that consisted of four main steps:

1. Manually aligning the LUs from RBN and DWN: mapping LUs to synsets, splitting, merging, deleting LUs and/or synsets
2. Adding essential information to new LUs: combinatorics, definitions, examples, etc.
3. Adding essential information to new synsets: semantic relations, SUMO mappings, Princeton Wordnet2.0 mappings
4. Manually verifying or creating mappings from existing synsets to Princeton Wordnet2.0, SUMO and WordNet Domains

The semantic information in the LUs and the synsets is complementary. As an example of the complementary combination of information, we discuss the verb *zetten* (to prepare). For the preparation of food and drinks normally the verb *maken* is used (*limonade maken* to make lemonade). This information can be found in the synset. However, in the case of making coffee or tea, one should use the verb *zetten*. The lexical constraints on phrasing the relations are not in the synsets but are provided by the LUs. Occasionally, mapping LUs and synsets raised some fundamental semantic questions. An example is the LU *brouwen* (to brew beer). This single LU corresponds to three synsets, meaning 'to brew', 'preparing a meal' and 'making plans'. The two additional meanings in DWN are metaphorical extensions; *brouwen* goes with the association of preparing, making or inventing something in an obscure way. In the synset for the concept of preparing a meal, *brouwen* is the only synonym with a clear-cut negative association. The synonyms *klaarmaken*, *toebereiden*, *bereiden* (to prepare, to make) and *koken* (to cook) are

[9]Note that the number of senses in the Cornetto database may be different from the original RBN and DWN. The RBN-based sense sequences are mostly the same, DWN-based sense sequences are mostly different.

neutral. This problem shows that the LUs and synsets differ in their perspective on word meaning. From the perspective of a LU, the aspects of meaning shared by a set of synonyms is not always an obvious meaning of a word form. As a result, aligning LUs and synsets sometimes leads to problems. In the LU of *brouwen*, the metaphorical meaning of preparing a meal was added for making the alignment with the synset possible.

In total over 10K LUs have been edited manually, corresponding to about 4,500 words that represent the most polysemous and most frequent words in the database. Another set of 509 nouns having 8 or more equivalences to PWN and 618 verbs with 5 or more equivalences were manually revised in terms of step 4. If synsets got too many mappings through the automatic mapping software, the relations are usually of low-quality and therefore also the import of the SUMO and WordNet Domain labels is unreliable.

10.6 Qualitative and Quantitative Results

In this section, an overview is presented of the main results of the alignment of DWN and RBN. First, we provide an overview of the size and coverage of the Cornetto database. Next, we discuss the quality of the alignment. Finally, we report on two task-based evaluations of the database.

Table 10.2 gives some overview statistics on the size and coverage of the database. Cornetto has about 1.6 more synsets and 1.4 as many word meanings as in the original Dutch wordnet. Our coverage compared to Princeton Wordnet is about 60 %. The average polysemy is 1.07 for nouns, 1.56 for verbs and 1.05 for adjectives.The average synset size is 1.47. The main statistics for the top-level elements of the synset data are given: the total number of synonyms, the number of internal semantic relations (like hypernyms, antonyms, meronyms, etc.), the equivalence relations to Princeton Wordnet, Wordnet Domain mappings, and SUMO mappings. Many relations are one-to-many thus exceeding the number of synsets. Furthermore, almost half of the synsets have definitions, which are derived from the resume fields of all the LUs that are synonyms of a synset.

Table 10.3 gives the main statistics for the combinatorial information related to the LUs. The 85,418 examples are subdivided into different categories: free examples illustrate the use of a LU in a wide context: the fixed examples include lexical collocations, i.e. frequent combinations with other nouns, verbs and adjectives; the grammatical collocations are frequent combinations with function words. Further, pragmatic collocations provides expressions which are associated with a fixed communicative situation.

Additional labels for the quality rates for the LU-synset mappings have been stored in a separate database. If the LU-to-synset alignment was checked manually, the quality is 100 % (10,120 alignments (9.86 %)) . These are all high frequent and high polysemous verbs, nouns and adjectives. If the mapping was not checked manually, the quality rates depend on the heuristics that underlie the mapping.

Table 10.2 Overview data Cornetto repositories

–	All	Nouns	Verbs	Adjectives
Synsets	70,370	52,845	9,017	7,689
Lexical units	119,108	85,449	17,314	15,712
Lemmas (form+POS)	92,686	70,315	9,051	12,288
Synonyms in synsets	103,762	75,475	14,138	12,914
Synonyms per synset	1.47	1.43	1.57	1.68
Senses per lemma	1.12	1.07	1.56	1.05
Definitions	35,769	25,460	6,157	4,152
Semantic relations	89,934	68,034	15,811	6,089
Equivalence relations	85,293	53,974	13,916	17,403
Domain relations	93,419	70,522	11,073	11,824
SUMO relations	70,002	46,964	12,465	10,573

Table 10.3 Overview of combinatorial information for LUs

–	All	Nouns	Verbs	Adjectives
Free examples	44,669	18,242	12,565	13,862
Lexical collocations	19,173	17,282	784	1,107
Grammatical collocations	10,407	6,869	3,160	378
Pragmatic collocations	1,472	637	565	270
Idioms	9,365	6,893	1,416	1,056
Proverbs	332	157	102	73
Total	85,418	50,080	18,592	16,746

Table 10.4 provides an overview of the quality labels that have been assigned. The number suffix after the labels indicates the reliability of the heuristic, based on a sample of 100 records per part-of-speech. For example, M-97 is a mapping with a confidence of 97 %. We see that about 53 % of the mapping records have no value for status. This means that none of the editors checked the matches and that they have not been validated in a post selection. Most of these are low frequent nouns that only occur in DWN with no match in RBN.

Additionally, two small task-based evaluations were carried out to assess the added value of the combined databases: classifying news and bootstrapping a subjectivity lexicon.

In the first task, we investigated if word combinations from Cornetto provide strong triggers for the classification of news articles using topic labels such as *sports*, *economy*, etc. [8]. The word combinations consist of lemmas combined with content words from the definitions, the examples and related synonyms and synsets. For the word *band*, we would extract combination such as *band-muziek*, *band-oppompen*, *band-moederband*. We thus extracted 60,262 records for unique forms with 396,348 word combinations. When processing news articles, the system checks every content word in the text to see if there was another content word close to it (in a window of ten words) that forms a Cornetto combination. If so, the program adds the individual content words to the index but also the combination. The assumption is that the

Table 10.4 Quality labels for *automatic* alignments

Quality label	Total	Percentage (%)	Specification
M-97	25,234	24.13	Monosemous words
B-95	4,944	4.73	Bisemous words. The first sense and second sense of RBN are aligned with the first and second sense of DWN
BM-90	4,214	4.03	Words that have one sense in RBN and two senses in DWN or vice versa. The first sense is aligned
Resume-75	1,047	1.00	Alignments based upon a substantial overlap between the RBN and DWN definitions and synonyms
D-75	2,085	1.99	Alignments of nouns that have an automatic alignment score higher than 30 %
D-58	774	0.74	Alignments of verbs that have an automatic alignment score higher than 30 %
D-55	171	0.16	Alignments of adjectives that have an automatic alignment score higher than 30 %
No-status	55,975	53.53	

combinations have a higher information value for the text than the individual words, which may be ambiguous. The baseline system was trained in a the classical way with the separate words only. As an alternative system, we also built indexes with all bigrams occurring in the training texts in addition to the individual words. For evaluation, 40 manually classified test documents were processed in the same way as the text for each of the three classification systems. The Cornetto combinations resulted in 2.8 % higher F-measure, 4.5 % higher recall and 0.5 % higher precision than the baseline. At the same time the indexes were only 1.2 times bigger by indexing word combinations using Cornetto. This means that it presents a realistic technology enhancement. The bigram system performed lower than the baseline, while its index is the biggest (4.6 times the baseline index).

The second task-based evaluation presented an algorithm that bootstraps a subjectivity lexicon from a list of initial seed examples [9]. The algorithm considers a wordnet as a graph structure where similar concepts are connected by relations such as synonymy, hyponymy, etc. We initialised the algorithm by assigning high weights to positive seed examples and low weights to negative seed examples. These weights are then propagated through the wordnet graph via the relations in the graph. After a specified number of iterations, words are ranked according to their weight. Words at the top of this ranked list are assumed to be positive and words at the bottom of the list are assumed to be negative. The algorithm was implemented and ran using two different wordnets available for Dutch: the original DWN and Cornetto. We found that using Cornetto instead of DWN resulted in a 7 % improvement of classification accuracy in the top-1500 positive words and in the top-2000 negative words. Between 70–86 % of this improvement can be attributed to the larger size of Cornetto, the remaining improvement is attributed to the larger set of relations between words.

10.7 Acquisition Toolkits

In addition to the Cornetto database, two acquisition toolkits have been developed to enhance automatic extension of the database. The first toolkit focused on automatic extraction of hypernym pairs from the web and a newspaper corpus. The second module was designed to extract domain-specific terms and collocations.

10.7.1 Acquisition Toolkit for Hypernym Pairs

For this toolkit, methods for improving the coverage of the lexical database were examined [27]. In particular, we evaluated methods for automatically extracting hypernym pairs. In texts, evidence for such a relation can be found in fixed word patterns. For example Hearst [17] explored using text patterns for finding hypernym pairs. We used the extraction approach outlined in [26] for automatically finding word pairs related by hypernymy. We extracted word pairs from the Dutch part of EuroWordNet [32] and used these as examples to train a machine learner for identifying interesting text patterns in which the pairs occurred. Next, we used the patterns that were found by the machine learner to identify other word pairs that could be related by hypernymy. We used the same machine learner as [26]: Bayesian Logistic Regression [15].

We evaluated the performance of individual text patterns and combinations of patterns on the task of extracting hypernym pairs from text. We applied the extraction method both to texts from a newspaper corpus and web text, and compared the approaches to a morphological baseline which stated that every complex noun has its final part as a hypernym, which for example predicts *bird* as hypernym of *blackbird*. We found that combined patterns outperformed individual patterns and the large web corpus outperformed the newspaper corpus. However, to our surprise none of the extraction techniques outperformed the baseline [27].

We provided the results of the newspaper texts in an online web demo [28]. The precision of the results (31 %) is in line with the state of the art, but not good enough to be useful for automatic extension of the Cornetto database. As a result , the output of the acquisition tool was not used in the construction phase of Cornetto.

10.7.2 Acquisition Toolkit for Specific Domains

In addition, we have developed a toolkit for the creation of a domain-specific lexicon containing terms and collocations [4]. For the identification of domain-specific terms and collocations, we assume large text corpora from which the terms are learned by means of statistical methods. We have experimented with common association metrics such as the likelihood ratio for a binomial distribution and a chi-square statistic, and with frequent item set mining.

The toolkit was evaluated on texts of the medical and legal domains written in Dutch. The corpora regard the medical texts of the Merck Manual, the medical encyclopedia from Spectrum and the Dutch Wikipedia articles classified in the category medicine. We created a financial law corpus obtained from the EURLex collection. As a general corpus we considered the Dutch Wikipedia pages.

For the recognition of domain-specific terms, where the association between the occurrence of a term and a domain-specific corpus was measured, the best results were obtained with a chi-square metric. Due to the lack of large and suitable training data, the results in terms of the F-1 measure hardly pass 30 %, when the extracted terms are compared with the terms found in an online medical lexicon .

Collocations can be defined as a combination of words that occur in a certain rigid order. We have extracted multi-word units, ranging from free word combinations that often co-occur to fixed idioms. The extraction is done in two steps. First, candidate collocations are identified. Second, the obtained candidates are filtered by imposing a syntactic template. In this setting, collocations could be detected such as *interne markt* (internal market), when imposing the adjectivenoun constraint, *rekening houden met* (taking into account) when imposing the noun-verb-preposition constraint, and *artikel van verordering* (article of regulation) when imposing the noun-preposition-noun constraint. According to a limited manual inspection, the obtained collocations are of good quality and the best results were obtained by frequent item set mining. Unfortunately, a complete formal evaluation by domain experts has never been performed.

10.8 Further Development of Cornetto

A number of subsequent projects have been launched that build on Cornetto:

- DutchSemCor[10] creates a sense-tagged corpus and word-sense-disambiguation software. Within this project, the Cornetto database is also extended with new, corpus-based word meanings, example sentences and semantic relations [34].
- Cornetto-LMF-RDF converts an updated version of the Cornetto database into the ISO standard LMF and the W3C standard RDF.
- Europeana[11] is a search portal on museum archives that uses the Cornetto database to provide Dutch-English cross-lingual search on meta data. Searching for *window* likewise gives results for Dutch *venster*.
- In the context of the Europeana project[12], the DWN part of Cornetto was also made available as an RDF file that consists of 792,747 triples. Cornetto-RDF is published in the linked open data cloud and linked to Wordnet W3C.

[10] www2.let.vu.nl/oz/cltl/dutchsemcor/

[11] eculture.cs.vu.nl/europeana/session/search

[12] www.europeana.eu

- FromTextToPoliticalPositions[13] develops an extension of Cornetto with fine-grained subjectivity features used for extracting political positions from text.
- SemanticsOfHistory[14] develops a domain-specific extension of Cornetto to mine historical events from text that are used in the CATCH project Agora[15].
- KYOTO[16] developed a generic fact mining platform using Cornetto as a resource. Within this project, an additional set of mappings to Princeton Wordnet was updated and edited.
- Daeso[17] is another STEVIN project that measures similarity across text. It implemented some state-of-the-art similarity measures developed for the Princeton Wordnet on top of the Cornetto database. Daeso also created a python client that can access the Cornetto database.

Since the release of Cornetto in 2008, 21 licenses have been issued by the HLT Agency in a period of 3 years.

10.9 Conclusion

The Cornetto project created a unique semantic database for Dutch and for the language community at large. The aligned information from two previously unconnected lexical databases provides a very rich database with semantic relations between concepts and traditional lexicographic information about the lexical units: e.g. combinatorics, collocations and pragmatics. By maintaining two separate collections, Cornetto provides two different views on the semantic organisation of the lexicon, which provides a form basis for studying semantics of Dutch and for developing language-technology applications. Alignment of two very differently organised lexicons proved feasible, however we argue that manual checks and editing are necessary to improve the overall quality and to solve semantic issues that stem from the different structures of the lexicons. Furthermore, the automatic acquisition toolkits provided some promising results, but also showed that acquiring a semantic lexicon from natural texts is extremely difficult for high-frequent and polysemous words and is hampered by some constraints. For instance, relations that hold between concepts are often not expressed in text as these relations are obvious for a reader.

Another major contribution is the mapping to the SUMO ontology, which allows us to differentiate rigid from non-rigid concepts and clarify the relations to entities and processes. This was taken up in subsequent projects such as KYOTO and

[13] www2.let.vu.nl/oz/cltl/t2pp/

[14] www2.let.vu.nl/oz/cltl/semhis/index.html

[15] agora.cs.vu.nl/

[16] www.kyoto-project.eu/

[17] daeso.uvt.nl/

the Global Wordnet Grid. This provides a first fundamental step towards a further formalisation of the semantics of the Dutch language and the possibility to develop semantic web applications. There is a plethora of possibilities to further extend and enrich the Cornetto database. We are considering mappings to FrameNet and creating mappings from multiword units and idioms to synsets, as well as the development of WSD systems that can assign Cornetto word meanings to words in contexts. A new version of the Cornetto database is scheduled for 2012 and includes revisions made during the DutchSemCor project [34].

References

1. Ageno A., Ribas F., Rigau G., Rodríguez H., Samiotou A.: TGE: tlinks generation environment. Proceedings of COLING'94. Kyoto, Japan (1994)
2. Atserias J., Climent S., Farreres J., Rigau G., Rodríguez H.: Combining multiple methods for the automatic construction of multilingual WordNets. Proceedings of RANLP'97. Tzigov Chark, Bulgaria (1997)
3. Baker, C., Fillmore, C., Lowe, J.: The Berkeley Framenet project. Proceedings of COLING/ACL98, Montreal, Canada (1998)
4. Boiy, E., Moens, M.-F.: Extracting domain specific collocations for the Dutch WordNet. Technical Report, Computer Science, K.U.Leuven (2008)
5. Chan, D.K., Wu, D.: Automatically merging lexicons that have incompatible part-of-speech categories. Joint SIGDAT Conference (EMNLP/VLC-99), Maryland (1999)
6. Copestake A., Briscoe E., Vossen P., Ageno A., Castelln I., Ribas F., Rigau G., Rodríguez, H., Samiotou, A.: Acquisition of lexical translation relations from MRDs. Mach Trans **9**, 3,33–3,69.
7. Cornetto deliverable D2 www2.let.vu.nl/oz/cltl/cornetto/docs/D02_Alignment of the Dutch databases in Cornetto.pdf
8. Cornetto deliverable D13 www2.let.vu.nl/oz/cltl/cornetto/docs/D03_Top-level ontology, relation constraints.pdf
9. Cornetto deliverable D14 www2.let.vu.nl/oz/cltl/cornetto/docs/D14 Tasked based evaluation subjectivity lexicon.pdf
10. Cornetto deliverable D15 www2.let.vu.nl/oz/cltl/cornetto/docs/D15_EvaluationReportCornetto.pdf
11. Cornetto deliverable D16 www2.let.vu.nl/oz/cltl/cornetto/docs/D16_Cornetto documentation (V12) v7.pdf
12. Crouch, D., King, T.H.: Unifying lexical resources. Proceedings of Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes. Saarbrücken, Germany (2005)
13. Farreres, X., Rigau, G., Rodríguez, H.: Using WordNet for building WordNets. Proceedings of COLING-ACL Workshop "Usage of WordNet in Natural Language Processing Systems", Montreal, Canada (1998)
14. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT, Cambridge (1998)
15. Genkin, A., Lewis, D., Madigan, D.: Large-Scale Bayesian Logistic Regression for Text Categorization. Technical report, Rutgers University, New Jersey (2004)

16. Guarino, N., Welty, C. Identity and subsumption. Green, R., Bean, C., Myaeng, S. (eds.), The Semantics of Relationships: An Interdisciplinary Perspective. Kluwer, Dordrecht, The Netherlands (2002)
17. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. Proceedings of ACL-92, Newark, DE, USA (1992)
18. Horák, A., Pala, K., Rambousek, A., Povolný, M.: DEBVisDic First Version of New Client-Server Wordnet Browsing and Editing Tool. Proceedings of GWC-06. Jeju Island, Korea (2006)
19. Laparra E., Rigau, G., Cuadros, M.: Exploring the integration of WordNet and FrameNet Proceedings of GWC2010, Mumbai, India, January 31- February 4, 2010.
20. Maks, I., Martin, W., de Meerseman, H.: RBN Manual, Vrije Universiteit, intern publication, Amsterdam (1999)
21. Martin, W.: Referentiebestand Nederlands : Documentatie Vrije Universiteit Amsterdam Internal publication (2005). Productcatalogus/RBN. www.tst.inl.nl
22. Magnini, B., Cavagliá, G.: Integrating subject field codes into WordNet. Proceedings of the LREC, Athens, Greece (2000)
23. Molinero, M.A., Sagot, B., Lionel, N.: Building a morphological and syntactic lexicon by merging various linguistic resources. Proceedings of the NODALIDA-09, Danemar (2009)
24. Niles, I., Pease, A.: Mapping WordNet to the suggested upper merged ontology. Proceedings of the IKE'03, Las Vegas, NV, USA (2003)
25. Padr, M., Bel, N., Necsulescu, S.: Towards the automatic merging of lexical resources. Proceedings of the RANLP, Hisar, Bulgaria (2011)
26. Snow, R., Jurafsky, D., Ng, A.Y.: Semantic taxonomy induction from heterogenous evidence. Proceedings of the COLING/ACL 2006, Sydney, NSW, Australia (2006)
27. Tjong Kim Sang, E., Hofmann, K.: Automatic extraction of Dutch hypernym-hyponym pairs. Proceedings of the CLIN-2006, Leuven (2007)
28. Tjong Kim Sang, E.: Cornetto Dutch Set Demo. Online web demo (2007). www.let.rug.nl/erikt/bin/setdemo.cgi
29. Soria, C., Monachini, M., Vossen, P.: Wordnet-LMf: fleshing out a standardized format for wordnet interoperability Proceedings of the IWIC2009, Stanford, CA, USA (2009)
30. Toral A., Monachini M., Soria C., Cuadros M., Rigau G., Bosma, W., Vossen, P.: Linking a domain thesaurus to WordNet and conversion to WordNet-LMF. Proceedings of the ICGL 2010, Hong Kong, China (2010)
31. Van Hage, W.: Evaluating ontology alignment techniques PhD Thesis, VU University Amsterdam (2008)
32. Vossen, P. (ed.): EuroWordNet: a multilingual database with lexical semantic networks for European Languages. Kluwer, Dordrecht (1998)
33. Vossen, P. (ed.): EuroWordNet General Document1, Version 3 (Final) , University of Amsterdam (2002). http://www.vossen.info/docs/2002/EWNGeneral.pdf
34. Vossen P., Gorog, A., Laan, F., Van Gompel, M., Izquierdo, R., van den Bosch, A.: DutchSemCor: building a semantically annotated corpus for Dutch. Proceedings of the eLEX2011, Bled, Slovenia, November 10–12, 2011

Chapter 11
Dutch Parallel Corpus: A Balanced Parallel Corpus for Dutch-English and Dutch-French

Hans Paulussen, Lieve Macken, Willy Vandeweghe, and Piet Desmet

11.1 Introduction

Parallel corpora are a valuable resource for researchers across a wide range of disciplines, i.e. machine translation, computer-assisted translation, terminology extraction, computer-assisted language learning, contrastive linguistics and translation studies. Since the development of a high-quality parallel corpus is a time-consuming and costly process, the DPC project aimed at the creation of a multifunctional resource that satisfies the needs of this diverse group of disciplines.

The resulting corpus—the Dutch Parallel Corpus (DPC)—is a ten-million-word, sentence-aligned, linguistically enriched parallel corpus for the language pairs Dutch-English and Dutch-French. As the DPC is bidirectional, the corpus can also be used as comparable corpus to study the differences between translated versus non-translated language. A small part of the corpus is trilingual. The DPC distinguishes itself from other parallel corpora by having a balanced composition (both in terms of text types and translation directions), by its availability to the wide research community thanks to its copyright clearance and by focusing on quality rather than quantity.

To guarantee the quality of the text samples, most of them were taken from published materials or from companies or institutions working with a professional translation division. Care was taken to differentiate kinds of data providers, among them providers from publishing houses, press, government, corporate enterprises, European institutions, etc. To guarantee the quality during data processing, 10 %

H. Paulussen (✉) · P. Desmet
ITEC-IBBT KULeuven Kulak, Kortrijk, Belgium
e-mail: hans.paulussen@kuleuven-kulak.be; piet.desmet@kuleuven-kulak.be

L. Macken · W. Vandeweghe
Language and Translation Technology Team (LT^3), University College Ghent and Ghent University, Gent, Belgium
e-mail: lieve.macken@hogent.be; willy.vandeweghe@hogent.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_11, © The Author(s) 2013

of the corpus has been manually verified at different levels, including sentence splitting, alignment and linguistic annotation. On the basis of these manually verified data, spot-checking and automatic control procedures were developed to verify the rest of the corpus. Each sample in the corpus has an accompanying metadata file. The metadata will enable the corpus users to select the texts that fulfil their specific requirements. The entire corpus is released as full texts in XML format and is also available via a web interface, which supports basic and complex search queries and presents the results as (parallel) concordances.

The remainder of this paper is structured as follows. Section 11.2 focuses on corpus design and data acquisition, while Sect. 11.3 elaborates on the different corpus processing stages. Section 11.4 contains the description of the two DPC exploitation formats along with the first exploitation results of the corpus in different research domains. Section 11.5 ends with some concluding remarks.

11.2 Corpus Design and Data Acquisition

The design principles of DPC were based on research into standards for other parallel corpus projects and a user requirements study. Three objectives were of paramount importance: balance (Sect. 11.2.1), quality of the text samples and IPR clearance (Sect. 11.2.2).

11.2.1 Balanced Corpus Design

The Dutch Parallel Corpus consists of two language pairs (Dutch-English and Dutch-French), has four translation directions (Dutch into English, English into Dutch, Dutch into French and French into Dutch) and five text types (administrative texts, instructive texts, literature, journalistic texts and texts for external communication). The DPC is balanced both in terms of text types and translation directions.

In order to enhance the navigability of the corpus, a subdivision was imposed on the five text types resulting in the creation of a finer tree-like structure within each type. This subdivision has no implications for the balancing of the corpus. The introduction of subtypes is merely a way of mapping the actual landscape within each text type, and assigning accurate labels to the data in order to enable the user to correctly select documents and search the corpus. A division could also be made between two main data sources: commercial publishers versus institutions and companies (cf. Table 11.1). For a detailed description of the DPC corpus design and text typology, we refer to [17, 24].

All information on translation direction and text types has been stored in the metadata files, complemented with other translation- and text-related information such as the intended audience, text provider, etc.

Table 11.1 DPC text types and subtypes according to data source

Source	Text type	Subtype
Institutions / Companies	Administrative texts	Legislation
		Proceedings of debates
		Minutes of meetings
		Yearly reports
		Official speeches
	External communication	(Self-)presentation
		Informative documents
		Promotion/advertising
		Press releases
		Scientific texts
	Instructive texts	Manuals
		Legal documents
		Procedure descriptions
Publishers	Journalistic texts	News reporting articles
		Comment articles
	Literature	Novels
		Essayistic texts
		(Auto)biographies
		Expository works

The Dutch Parallel Corpus consists of more than ten million words, distributed over five text types, containing 2,000,000 words each. Within each text type, each translation direction contains 500,000 words. In order to preserve a good balance, the material of each cell (i.e. the unique combination of text type and translation direction) originates from at least three different providers. The exact number of words in DPC can be found in Table 11.2.[1] When compiling DPC, we were forced to make two exceptions to the global design:

- Given the difficulty to find information on translation direction for instructive texts, the condition on translation direction was relaxed for this text type.
- For literary texts, it often proved difficult to obtain copyright clearance. For that reason, the literary texts are not strictly balanced according to translation direction, but are balanced according to language pair.

The creation of a corpus that is balanced both in terms of text types and translation directions relies on a rigorous data collection process, basically consisting of two phases:

- Finding text providers who offer high-quality text material in accordance with the design prerequisites and convincing them to participate in the project.
- Clearing copyright issues for all the texts that are integrated in the corpus.

[1]The word counts are all based on clean text, meaning that all figures, tables and graphs were removed. "X" stands for unknown source language.

Table 11.2 DPC word counts per text type and translation direction

Text type	SRC ⇒ TGT	DU	EN	FR	Total
Administrative texts	EN ⇒ DU	255,155	246,137		501,292
	FR ⇒ DU	307,886		322,438	630,324
	DU ⇒ EN	249,410	257,087		506,497
	DU ⇒ FR	280,584		301,270	581,854
	Total	1,093,035	503,224	623,708	2,219,961
External communication	EN ⇒ DU	278,515	272,460		550,975
	FR ⇒ DU	233,277		250,604	483,881
	DU ⇒ EN	246,448	255,634		502,082
	DU ⇒ FR	241,323		270,074	511,397
	X ⇒ D/E	21,679	20,118		41,797
	X ⇒ D/E/F	14,192	14,953	15,743	44,888
	Total	1,035,434	563,165	536,421	2,135,020
Instructive texts	EN ⇒ DU	340,097	327,543		667,640
	FR ⇒ DU	40,487		42,017	82,504
	DU ⇒ EN	19,011	20,696		39,707
	DU ⇒ FR	110,278		115,034	225,312
	X ⇒ D/F	59,791		73,758	133,549
	X ⇒ D/E	299,996	296,698		596,694
	X ⇒ D/E/F	138,673	145,103	166,836	450,612
	Total	1,008,333	790,040	397,645	2,196,018
Journalistic texts	EN ⇒ DU	262,768	264,900		527,668
	FR ⇒ DU	240,785		265,530	506,315
	DU ⇒ EN	250,580	259,764		510,344
	DU ⇒ FR	314,989		340,319	655,308
	Total	1,069,122	524,664	605,849	2,199,635
Literature	EN ⇒ DU	148,488	143,185		291,673
	FR ⇒ DU	186,799		186,620	373,419
	DU ⇒ EN	346,802	361,140		707,942
	DU ⇒ FR	323,158		348,343	671,501
	Total	1,005,247	504,325	534,963	2,044,535
Grand total		5,211,171	2,885,418	2,698,586	10,795,175

11.2.2 Data Collection and IPR

An ideal data collection process consists of three or maybe four steps: a researcher finds adequate text material that should be included in the corpus, he/she contacts the legitimate author and asks his/her permission, the author agrees and both parties sign an agreement. As experienced during the whole project period, this process

is in reality far more complicated[2] and negotiations lasting 1–2 years were not exceptional.

As was briefly mentioned before, two main data sources can be distinguished on the basis of text provider type, namely commercial publishers versus institutions and companies. This main distinction can be considered as an anticipator on the difficulties encountered during data collection. When text production is a text provider's core business (e.g. newspaper concerns, publishing agencies, etc.), one can intuitively expect longer negotiation cycles.

Throughout the project period, clearing copyright issues proved a difficult and time-consuming task. For all IPR matters, the DPC team worked in close collaboration with the HLT agency that drew up the agreement templates.

Due to the heterogeneity of text providers (55 text providers donated texts to DPC) different types of IPR agreements were made: a standard IPR agreement, an IPR agreement for publishers, a short IPR agreement and an e-mail or letter with permission. Although specific changes often had to be made in the agreements, all texts included in the corpus were cleared from copyrights at the end of the project period. Using different agreements was a great help in managing negotiations with text providers and bringing them to a favourable conclusion. For a detailed description of data collection, IPR agreements, practical guidelines and advice, we refer to [7].

11.3 Corpus Processing

After collecting the different texts and normalizing the format, the actual processing of the corpus can start. The main task consisted in aligning the texts at sentence level (Sect. 11.3.1). The second task involved an extra layer of linguistic annotation: all words were lemmatized and grammatically tagged (Sect. 11.3.2).

The different processing stages were carried out automatically. For reasons of quality assurance, each processing stage was checked manually for 10 % of the corpus. For the other part, spot-checking and automatic control procedures were developed.

11.3.1 Alignment

The main purpose of aligning a parallel corpus is to facilitate bilingual searches. Whereas in monolingual corpora you look for a word or a series of words, in a parallel corpus you also want to retrieve the corresponding words in the other language. This kind of search is only possible if the corpus is structured in such a way that all corresponding items are aligned. During alignment a particular text

[2] In the case of a parallel corpus more parties are involved: author, translator, publisher, and foreign publisher.

chunk (e.g. a sentence) in one language is linked to the corresponding text chunk(s) in the other language. The following alignment links are used within the DPC: 1:1, 1:many, many:1, many:many, 0:1 and 1:0. Many-to-many alignments are used in the case of overlapping or crossing alignments. Zero alignments occur when no translation could be found for a sentence in either the source or the target language.

In general, there are two types of alignment algorithms: those based on sentence-length and those based on word correspondence. Very often a mixture of the two is used. The two types differ mainly in the method used: a statistical vs. a heuristic method [22]. The first type starts from the assumption that translated sentences and their original are similar in length. The correspondence between these sentences is either expressed in number of words (for example Brown et al. [2]) or in number of characters per sentence (for example Gale and Church [11]). On the basis of probability measures, the most likely alignment is then selected.

The second type of algorithms starts from the assumption that if sentences are translations of one another, the corresponding words must be translations as well. In this lexical approach the similarity of translated words is calculated on the basis of specific associative measures. To determine the degree of similarity between translated words, an external lexicon can be used, or a translation lexicon can be derived from the texts to be aligned [13]. In a more linguistic approach, one could look for morphologically related words or *cognates*, which can be very helpful for languages having similar word forms, as is the case for English and French [26].

Three different alignment tools were used to align all sentences of DPC, each of them having particular advantages and drawbacks.

The Vanilla Aligner developed by Danielsson and Ridings [6] is an implementation of the sentence-length-based algorithm of Gale and Church [11]. This tool aligns sentences within blocks of paragraphs, and therefore requires the same number of paragraphs for both languages, which can be a limitation, since the slightest shift in number of paragraphs blocks the whole alignment process. Therefore, in the DPC project, paragraph alignment has been carried out prior to sentence alignment by adopting a very pragmatic approach: only if the number of paragraphs or the size of the paragraphs differed, paragraph alignment was manually verified.

The Geometric Mapping and Alignment (GMA) developed by Melamed [19] uses a hybrid approach, based on word correspondences and sentence length. The system looks for cognates and can make use of external translation lexicons. The DPC project made use of the NL-Translex translation lexicons [12] as additional resources for recognizing word correspondences.

The Microsoft Bilingual Aligner developed by Moore [21] uses a three-step hybrid approach involving sentence and word alignment. In the first step, a sentence-length-based alignment is established. The output of the first step is then used as the basis for training a statistical word alignment model [3]. In the final step, the initial set of aligned sentences is realigned using the information from the word alignments established in the second step. The quality of the aligner is very good, but the aligner outputs only 1:1 alignments, thus disregarding all other alignment types.

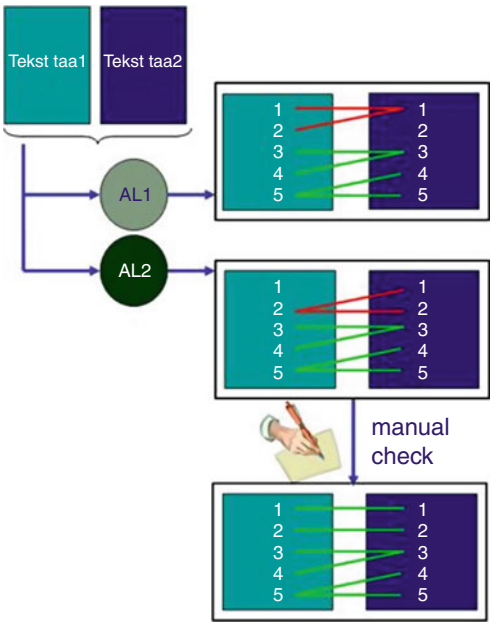


Fig. 11.1 Alignment spot check

Although each alignment tool has specific advantages and limitations, the combination of the three tools was a very helpful instrument in order to control the alignment quality of the DPC translations. Since the verification of a ten-million-word corpus is a time-consuming task, the manual verification could be limited to those cases where the three alignments diverged: when at least two aligners agreed, the alignment output could be considered of high quality. Thanks to this approach of alignment spot checks (cf. Fig. 11.1), only a small portion of the alignments was still to be checked by hand. More details on the performance of the different alignment tools used in the DPC project can be found in [15].

The entire corpus has been aligned at sentence level. The DPC also contains approximately 25,000 words of the Dutch-English part manually aligned at the sub-sentential level. These manually created reference alignments can be used to develop or test automatic word alignment systems. For more information on the sub-sentential alignments, we refer to [17].

11.3.2 Linguistic Annotation

Next to sentence alignment, the DPC data have been enriched with linguistic annotation, involving part-of-speech tagging and lemmatization to facilitate the linguistic exploration of any type of corpus. In the DPC project we have chosen to use annotation tools that are commonly available. In some cases, adaptation of the tools or pre-processing of the data was required.

Table 11.3 Performance of the PoS taggers and lemmatizers on a manually validated DPC sample

	Sample size (tokens)	Lemmata	PoS (full tag)	PoS (main category)
Dutch	211,000	96.5 %	94.8 %	97.4 %
English	300,000	98.1 %	96.2 %	N/A
French	330,000	98.1 %	94.6 %	97.4 %

For English, we opted for the combined memory-based PoS tagger/lemmatizer which is part of the MBSP tools set [5]. The English memory-based tagger was trained on data from the Wall Street Journal corpus in the Penn Treebank [18]. For Dutch, the D-Coi tagger was used [27], which is an ensemble tagger that combines the output of different machine learning algorithms. For French, we used an adapted version of TreeTagger [25].

The English PoS tagging process differs a lot from both Dutch and French grammatical annotation, in the sense that for the former a limited set of only 45 distinct tags is used, whereas both Dutch and French require a more detailed set of tags, because of their morpho-syntactic structure. In the case of Dutch, the CGN PoS tag set [28] was used, which covers word categories and subcategories, coding a wide range of morpho-syntactic features, thus amounting to a set of 315 tags. For French, we used the GRACE tag set which consists of 312 distinctive tags [23].

The tagging process for French required some adaptation of the tools, because the language model lacked lemmatized data, so that we were obliged to run the tool twice: first using the original parameter file, providing lemmata but containing only a limited tag set, and then using the enriched parameter file (provided by LIMSI [1]), containing the GRACE tag set but lacking lemmatized forms. Although the tagging process implied different processing steps, the result was also the basis for the spot check task. Similar to the alignment procedure, the combination of two annotation runs gave the necessary information to automatically detect which tags had to be verified manually. For example, if both tagging runs resulted in the same PoS tag, no further manual check was required.

The performance of the part-of-speech taggers and lemmatizers is presented in Table 11.3. The automatically predicted part-of-speech tags and lemmata were manually verified on approximately 800,000 words selected from different text types. For Dutch and French, both the accuracy score on the full tags (containing all morpho-syntactic subtags) and the score on the main tags are given. The obtained scores give an indication of the overall tagging accuracy that can be expected in DPC.

11.4 Corpus Exploitation

The final task of the DPC project consisted in packaging the data in such a way that the corpus can easily be exploited. In order to meet the requirements of different types of users, it was decided to make the corpus available in two different

Table 11.4 DPC filename patterns

Filename pattern	Description
`dpc-xxx-000000-nl-tei.xml`	Monolingual Dutch file
`dpc-xxx-000000-yy-tei.xml`	Monolingual English or French file
`dpc-xxx-000000-nl-mtd.xml`	Dutch metadata file
`dpc-xxx-000000-yy-mtd.xml`	English/French metadata file
`dpc-xxx-000000-nl-yy-tei.xml`	Alignment index file

formats. First of all, the corpus is distributed as a set of structured XML data files, which can be queried by any researcher acquainted with basic text processing skills (Sect. 11.4.1). On the other hand, a special parallel web concordancer was developed, which can easily be consulted over the internet (Sect. 11.4.2). This section describes both application modes and gives an overview of the first exploitation results of DPC (Sect. 11.4.3).

11.4.1 XML Packaging

The data have been packaged in XML in accordance with the TEI P5 standard. The choice for XML was motivated by the fact that it is a transparent format which can easily be transferred to other types of formats depending on the tools available to the developer. The XML files are well-formed and validated. The former is a basic requirement for XML files, whereas the latter gives more control over the structure of the XML files. Each XML file complies with the specifications of a basic TEI P5 DTD[3] stipulating, for example, that each word should contain attributes for part-of-speech and lemma.

For each language pair five different files are involved (cf. Table 11.4). First of all we have a text file for each language (e.g. dpc-xxx-000000-nl-tei.xml and dpc-xxx-000000-en-tei.xml representing a Dutch source file and an English target file). These data files contain the annotated sentences, where each word is grammatically tagged and lemmatized. To each data file a metadata file is linked (e.g. dpc-xxx-000000-nl-mtd.xml is the metadata file for dpc-xxx-000000-nl-tei.xml). Finally, an index file is used which contains all aligned sentences for the selected language pairs: for example, the index file dpc-xxx-000000-nl-yy-tei.xml contains all indexes for dpc-xxx-000000-nl-tei.xml and dpc-xxx-000000-en-tei.xml. The link between the different files is illustrated in Fig. 11.2.

Thanks to the validated XML format, it is possible to exploit the data files in different ways. A nice example is the development of the DPC web concordancing program—the second application mode of DPC—which is explained in the following section.

[3] A DTD (Document Type Definition) could be interpreted as a kind of text markup grammar, defining which markup elements can be used in which order.

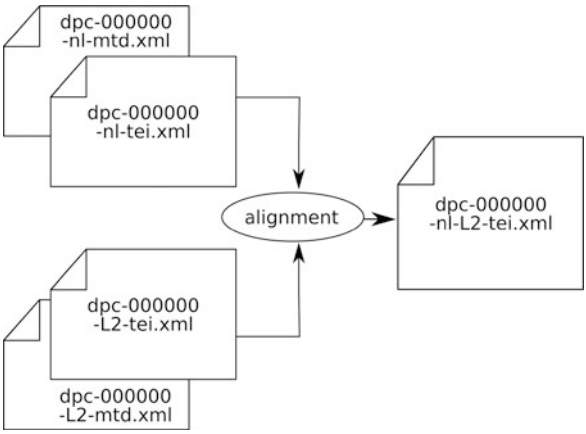


Fig. 11.2 DPC sentence-aligned files format

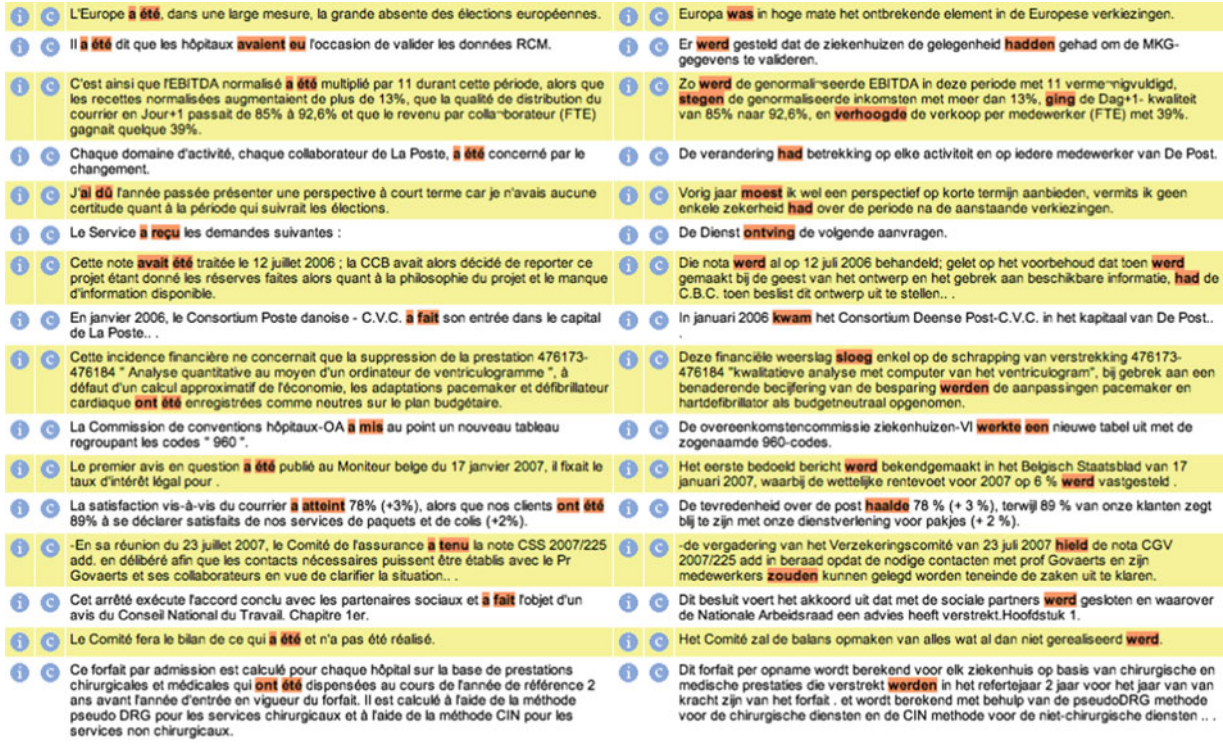
11.4.2 Parallel Web Concordance

A concordance program is a basic tool for searching a corpus for samples of particular words or patterns. Typically, the word or pattern looked for is presented in a context window, showing a certain number of context words left and right of the keyword. Therefore, such a concordancer is often called a KWIC-concordancer, referring to *keyword in context*. A parallel concordancer is a program written for displaying aligned data from a translation corpus. Since concordancers of this type are not as readily available as is the case with ordinary concordanders, and since they require a specific format, it was decided to develop a parallel concordancer especially for DPC.[4]

Parallel concordancers allows one to select words or patterns in one language and retrieve sample sentences from the selected language together with the corresponding aligned sentences in the other language. A better way consists in selecting words or patterns in the two languages. The DPC parallel concordancer is especially developed to make such an enriched bilingual search.[5] In Fig. 11.3 you can see the first output page of a combined query, which looks for French-Dutch text samples of the French *passé composé* matching the Dutch *verleden tijd* (simple past). The output is inevitably obscured by some noise—mainly due to complex sentence structure—, but the result is rich enough to allow researchers to further analyze the output, without having to call in the help of programmers. There is an exporting module to Excel, so that researchers can annotate the results in a more commonly used working format.

[4]The original web interface was developed by Geert Peeters and Serge Verlinde (ILT KU Leuven).

[5]A demo version of the parallel concordancer is available at the HLT agency via the following link: http://dpc.inl.nl/indexd.php



L'Europe **a été**, dans une large mesure, la grande absente des élections européennes.	Europa **was** in hoge mate het ontbrekende element in de Europese verkiezingen.
Il **a été** dit que les hôpitaux **avaient eu** l'occasion de valider les données RCM.	Er **werd** gesteld dat de ziekenhuizen de gelegenheid **hadden** gehad om de MKG-gegevens te valideren.
C'est ainsi que l'EBITDA normalisé **a été** multiplié par 11 durant cette période, alors que les recettes normalisées augmentaient de plus de 13%, que la qualité de distribution du courrier en Jour+1 passait de 85% à 92,6% et que le revenu par collaborateur (FTE) gagnait quelque 39%.	Zo **werd** de genormaliseerde EBITDA in deze periode met 11 vermenigvuldigd, **stegen** de genormaliseerde inkomsten met meer dan 13%, **ging** de Dag+1- kwaliteit van 85% naar 92,6%, en **verhoogde** de verkoop per medewerker (FTE) met 39%.
Chaque domaine d'activité, chaque collaborateur de La Poste, **a été** concerné par le changement.	De verandering **had** betrekking op elke activiteit en op iedere medewerker van De Post.
J'**ai dû** l'année passée présenter une perspective à court terme car je n'avais aucune certitude quant à la période qui suivrait les élections.	Vorig jaar **moest** ik wel een perspectief op korte termijn aanbieden, vermits ik geen enkele zekerheid **had** over de periode na de aanstaande verkiezingen.
Le Service **a reçu** les demandes suivantes :	De Dienst **ontving** de volgende aanvragen.
Cette note **avait été** traitée le 12 juillet 2006 ; la CCB avait alors décidé de reporter ce projet étant donné les réserves faites alors quant à la philosophie du projet et le manque d'information disponible.	Die nota **werd** al op 12 juli 2006 behandeld; gelet op het voorbehoud dat toen **werd** gemaakt bij de geest van het ontwerp en het gebrek aan beschikbare informatie, **had** de C.B.C. toen beslist dit ontwerp uit te stellen.. .
En janvier 2006, le Consortium Poste danoise - C.V.C. **a fait** son entrée dans le capital de La Poste.. .	In januari 2006 **kwam** het Consortium Deense Post-C.V.C. in het kapitaal van De Post.. .
Cette incidence financière ne concernait que la suppression de la prestation 476173-476184 " Analyse quantitative au moyen d'un ordinateur de ventriculogramme ", à défaut d'un calcul approximatif de l'économie, les adaptations pacemaker et défibrillateur cardiaque **ont été** enregistrées comme neutres sur le plan budgétaire.	Deze financiële weerslag **sloeg** enkel op de schrapping van verstrekking 476173-476184 "kwalitatieve analyse met computer van het ventriculogram", bij gebrek aan een benaderende becijfering van de besparing **werden** de aanpassingen pacemaker en hartdefibrillator als budgetneutraal opgenomen.
La Commission de conventions hôpitaux-OA **a mis** au point un nouveau tableau regroupant les codes " 960 ".	De overeenkomstencommissie ziekenhuizen-VI **werkte een** nieuwe tabel uit met de zogenaamde 960-codes.
Le premier avis en question **a été** publié au Moniteur belge du 17 janvier 2007, il fixait le taux d'intérêt légal pour .	Het eerste bedoeld bericht **werd** bekendgemaakt in het Belgisch Staatsblad van 17 januari 2007, waarbij de wettelijke rentevoet voor 2007 op 6 % **werd** vastgesteld .
La satisfaction vis-à-vis du courrier **a atteint** 78% (+3%), alors que nos clients **ont été** 89% à se déclarer satisfaits de nos services de paquets et de colis (+2%).	De tevredenheid over de post **haalde** 78 % (+ 3 %), terwijl 89 % van onze klanten zegt blij te zijn met onze dienstverlening voor pakjes (+ 2 %).
-En sa réunion du 23 juillet 2007, le Comité de l'assurance **a tenu** la note CSS 2007/225 add. en délibéré afin que les contacts nécessaires puissent être établis avec le Pr Govaerts et ses collaborateurs en vue de clarifier la situation... .	-de vergadering van het Verzekeringscomité van 23 juli 2007 **hield** de nota CGV 2007/225 add in beraad opdat de nodige contacten met prof Govaerts en zijn medewerkers **zouden** kunnen gelegd worden teneinde de zaken uit te klaren.
Cet arrêté exécute l'accord conclu avec les partenaires sociaux et **a fait** l'objet d'un avis du Conseil National du Travail. Chapitre 1er.	Dit besluit voert het akkoord uit dat met de sociale partners **werd** gesloten en waarover de Nationale Arbeidsraad een advies heeft verstrekt.Hoofdstuk 1.
Le Comité fera le bilan de ce qui **a été** et n'a pas été réalisé.	Het Comité zal de balans opmaken van alles wat al dan niet gerealiseerd **werd**.
Ce forfait par admission est calculé pour chaque hôpital sur la base de prestations chirurgicales et médicales qui **ont été** dispensées au cours de l'année de référence 2 ans avant l'année d'entrée en vigueur du forfait. Il est calculé à l'aide de la méthode pseudo DRG pour les services chirurgicaux et à l'aide de la méthode CIN pour les services non chirurgicaux.	Dit forfait per opname wordt berekend voor elk ziekenhuis op basis van chirurgische en medische prestaties die verstrekt **werden** in het refertejaar 2 jaar voor het jaar van van kracht zijn van het forfait . et wordt berekend met behulp van de pseudoDRG methode voor de chirurgische diensten en de CIN methode voor de niet-chirurgische diensten .. .

Fig. 11.3 Parallel concordancer output sample

Although it is possible to develop a full featured query interface, which allows for exploitation using regular patterns,[6] we have decided to restrict the interface to a small set of query patterns, transparent enough for non-experts to be able to find their way in exploring the parallel corpus without much hassle. Further exploitation is possible, if you analyse the XML source files, using XSLT or similar tools.

The DPC concordancer differs from similar parallel concordancers, in the sense that DPC has been provided with a extra annotation layers (PoS tags and lemmatization, and metadata), which allow for better selections, not possible in ParaConc or Multiconcord.[7] In the DPC concordancer, you can build subcorpora, based on metadata of text types and language filters. In the case of ParaConc, you cannot filter on extra annotation layers.

ParaConc and Multiconcord are platform specific. The first is available for Windows and Macintosh, the other only for Windows. The DPC concordancer is available over the internet and therefore not specifically linked to one platform. The DPC concordancer is freely available, but unlike the two others, adding new texts is not directly available.

11.4.3 First Exploitation Results of DPC

As mentioned in the introduction, it was the explicit aim of the DPC project to create a parallel corpus that satisfies the needs of a diverse user group. Since its (pre-)release DPC has been used in different research domains[8]:

- In the CAT domain, DPC has been used to select benchmarking data to evaluate different translation memory systems [14] and to extract language-pair specific translation patterns that are used in a chunk-based alignment system for English-Dutch [16].
- In the domain of CALL, DPC has been introduced as a valuable resource for language teaching. The corpus is being used as a sample repository for content developers preparing exercises for French and Dutch language learners [9]. Within CorpusCALL, parallel corpora like DPC are used as resources for data-driven language learning [20]. Parallel corpora are also useful instruments for rethinking the pedagogical grammaticography in function of frequency research. On the basis of such analysis one can find out, for example, how to teach the *subjonctif* for learners of French [29].

[6]Extended regular patterns are used in CQP (Corpus Query Processor) developed by IMS, and originally developed for CWB (Corpus Work Bench) (cf. also [4].)

[7]See http://www.athel.com/paraweb.pdf and http://artsweb.bham.ac.uk/pking/multiconc/lingua.htm for ParaConc and Multiconcord respectively.

[8]This is a non-exhaustive list as the authors are only aware of research making use of DPC conducted at their own institutions.

- In the framework of the DPC-project, a Gold Standard for terminology extraction was created. All terms (single- and multiword terms) were manually indicated in a set of texts belonging to two different domains (medical and financial). This Gold Standard has been used by Xplanation,[9] who as an industrial partner of the DPC project was partly responsible for the external validation of DPC.[10] It is also used in the TExSIS project[11] as benchmarking data to evaluate bilingual terminology extraction programmes.
- In the field of Translation Studies, DPC has been used as comparable corpus to study register variation in translated and non-translated Belgian Dutch [8] and [10]. More particulary, it was investigated to what extent the conservatism and normalization hypothesis holds in different registers of translated texts, compared to non-translated texts.
- Contrastive linguistics is another field where DPC has been used as a resource of authentic text samples. Vanderbauwhede [30, 31] studied the use of the demonstrative determiner in French and Dutch on the basis of corpus material from learner corpora and parallel corpora, including DPC. Although Dutch and French use the article and the demonstrative determiner in a quite similar way, parallel corpus evidence shows some subtle differences between both languages.

Furthermore, DPC is being used in a number of courses in CALL, translation studies and language technology. A substantial part of DPC has also been used for further syntactic annotation in the Lassy project (cf. Chap. 9, p. 147).

11.5 Conclusion

The DPC project resulted in a high-quality, well-balanced parallel corpus for Dutch, English and French.[12] Its results are available via the HLT Agency.[13] As part of the STEVIN objectives to produce qualitative resources for Dutch natural language processing, DPC is a parallel corpus that meets the requirements of the STEVIN programme.[14] The DPC corpus differs mainly from other parallel corpora in the following ways: (i) special attention has been paid to corpus design, which resulted in a well-balanced corpus, (ii) the corpus is sentence-aligned and linguistically annotated (PoS tagging and lemmatization), (iii) the different processing steps have been controlled in a systematic way and (iv) the corpus is available to the wide research community thanks to its copyright clearance.

[9] http://www.xplanation.com

[10] The Center for Sprogteknologi (CST) carried out a formal validation of DPC.

[11] http://lt3.hogent.be/en/projects/texsis/

[12] For further information, see the DPC project website: http://www.kuleuven-kortrijk.be/DPC

[13] http://www.tst-centrale.org

[14] A summary of the STEVIN requirements is given in the introduction of this book (cf. p. 1).

DPC is first of all used as a resource of translated texts for different types of applications, but also monolingual studies of Dutch, French and English can benefit greatly from it. The quality of the corpus—in content and structure—and the two application modes provided (XML and web interface) help to explain why the first exploitation results of DPC are promising.

References

1. Allauzen, A., Bonneau-Maynard, H.: Training and evaluation of POS taggers on the French MULTITAG corpus. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC-08), Marrakech, pp. 28–30 (2008)
2. Brown, P.F., Lai, J.C., Mercer, R.L.: Aligning sentences in parallel corpora. In: Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, pp. 169–176 (1991)
3. Brown, P.F., Della Pietra, V.J., Della Pietra, S.A., Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. **19**(2), 263–311 (1993)
4. Christ, O.: A modular and flexible architecture for an integrated corpus query system. In: Proceedings of COMPLEX, Conference on Computational Lexicography and Text Research, Budapest, pp. 23–32 (1994)
5. Daelemans, W., van den Bosch, A.: Memory-Based Language Processing. Cambridge University Press, Cambridge (2005)
6. Danielsson, P., Ridings, D.: Practical presentation of a vanilla aligner. In: Reyle, U., Rohrer, C. (eds.) The TELRI Workshop on Alignment and Exploitation of Texts, Ljubljana (1997)
7. De Clercq, O., Montero Perez, M.: Data collection and IPR in multilingual parallel corpora: Dutch parallel corpus. In: Proceedings of the 7th Language Resources and Evaluation Conference (LREC2010), Valletta, pp. 3383–3388 (2010)
8. Delaere, I., De Sutter, G., Plevoets, K.: Is translated language more standardized than non-translated language? Using profile-based correspondence analysis for measuring linguistic distances between language varieties. Target, **24**(2), (2012)
9. Desmet, P., Eggermont, C.: FRANEL: un environnement électronique d'apprentissage du français qui intègre des matériaux audio-visuels et qui est à la portée de tous. Cahiers F: revue de didactique français langue étrangère / Cahiers F: didactisch tijdschrift Frans vreemde taal pp. 39–54 (2006)
10. De Sutter, G., Delaere, I., Plevoets, K.: Lexical lectometry in corpus-based translation studies. Combining profile-based correspondence analysis and logistic regression modeling. In: Oakes, M., Ji, M. (eds.) Quantitative Methods in Corpus-Based Translation Studies. A Practical Guide To Descriptive Translation Research, pp. 325–345. John Benjamins, Amsterdam (2012)
11. Gale, W.A., Church, K.W.: A program for aligning sentences in bilingual corpora. Comput. Linguist. **19**(1), 75–102 (1993)
12. Goetschalckx, J., Cucchiarini, C., Van Hoorde, J.: Machine translation for Dutch: the NL-Translex project, Brussels/Den Haag, 16pp (2001)
13. Kay, M., Röscheisen, M.: Text-translation alignment. Comput. Linguist. **19**(1), 121–142 (1993)
14. Macken, L.: In search of the recurrent units of translation. In: Daelemans, W., Hoste, V. (eds.) Evaluation of Translation Technology. LANS 8/2009, pp. 195–212. Academic and Scientific Publishers, Brussels (2009)

15. Macken, L.: Sub-sentential alignment of translational correspondences. Ph.D. thesis, University of Antwerp (2010)
16. Macken, L., Daelemans, W.: A Chunk-Driven Bootstrapping Approach to Extracting Translation Patterns. In: Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics (Iasi, Romania). Lecture Notes in Computer Science, vol. 6009, pp. 394–405. Springer, Berlin/ Heidelberg (2010)
17. Macken, L., De Clerq, O., Paulussen, H.: Dutch parallel corpus: a balanced copyright-cleared parallel corpus. Meta **56**(2), 374–390 (2011)
18. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a Large Annotated Corpus of English: The Penn Treebank. Comput. Linguist. **19**(2), 313–330 (1993)
19. Melamed, D.I.: A portable algorithm for mapping bitext correspondence. In: Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics (ACL), Madrid, pp. 305–312 (1997)
20. Montero Perez, M., Paulussen, H., Macken, L., Desmet, P.: From input to output: the potential of parallel corpora for CALL. LRE (Submitted)
21. Moore, R.: Fast and Accurate Sentence Alignment of Bilingual Corpora. Machine Translation: From Research to Real Users. 2499, 135–144 (2002)
22. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29**, 19–51 (2003)
23. Paroubek, P.: Language resources as by-product of evaluation: the multitag example. In: Second International Conference on Language Resources and Evaluation (LREC 2000), Athens, pp. 151–154 (2000)
24. Rura, L., Vandeweghe, W., Montero Perez, M.: Designing a parallel corpus as a multifunctional translator's aid. In: Proceedings of XVIII FIT World Congress, Shangai, pp. 4–7 (2008)
25. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing, Manchester (1994)
26. Simard, M., Foster, G., Hannan, M.L., Macklovitch, E., Plamondon, P.: Bilingual text alignment: where do we draw the line? In: Botley, S., McEnery, A., Wilson, A. (eds.) Multilingual Corpora in Teaching and Research, pp. 38–64. Rodopi, Amsterdam (2000)
27. van den Bosch, A., Schuurman, I., Vandeghinste, V.: Transferring POS tagging and lemmatization tools from spoken to written Dutch corpus development. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), Genua (2006)
28. Van Eynde, F., Zavrel, J., Daelemans, W.: Part of speech tagging and lemmatisation for the spoken Dutch corpus. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000), Athens, pp. 1427–1434 (2000)
29. Van Keirsbilck, P., Lauwers, P., Desmet, P.: Le subjonctif tel qu'il s'enseigne en Flandre et en France: bilan et perspectives. Travaux de didactique du FLE. 64, 131–145 (2010)
30. Vanderbauwhede, G.: Le déterminant démonstratif en français et en néerlandais à travers les corpus: théorie, description, acquisition. Ph.D. thesis, K.U. Leuven (2011)
31. Vanderbauwhede, G.: Les emplois référentiels du SN démonstratif en français et en néerlandais: pas du pareil au même. J. Fr. Lang. Stud. **22**(2), 273–294 (2012) doi:10.1017/S0959269511000020

Chapter 12
Identification and Lexical Representation of Multiword Expressions

Jan Odijk

12.1 Introduction

Multi-word Expressions (MWEs) are word combinations with linguistic properties that cannot be predicted from the properties of the individual words or the way they have been combined. MWEs occur frequently and are usually highly domain-dependent. A proper treatment of MWEs is essential for the success of NLP-systems. This will be the topic of Sect. 12.2.

Generic NLP-systems usually perform less well on texts from specific domains. One of the reasons for this is clear: each domain uses its own vocabulary, and it uses generally occurring words with a highly specific meaning or in a domain-specific manner. For this reason, state-of-the-art NLP systems usually work best if they are adapted to a specific domain. It is therefore highly desirable to have technology that allows one to adapt an NLP system to a specific domain for MWEs, e.g., on the basis of a text corpus. Technology is needed that can identify MWEs in a maximally automated manner. This will be discussed in Sect. 12.3.

An NLP-system can only use an MWE if it is represented in a way suitable for that NLP system. Unfortunately, each NLP system requires its own formats and properties, and the ways MWEs are represented differs widely from NLP-system to NLP-system. Therefore a representation of MWEs that is as theory- and implementation-independent as possible and from which representations specific to a particular NLP system can be derived in a maximally automated manner is highly desirable. A specific approach to this, based on the Equivalence Class Method (ECM) approach, and applied to Dutch, will be described in Sect. 12.4.

J. Odijk (✉)
UiL-OTS, Trans 10, 3512 JK Utrecht, The Netherlands
e-mail: j.odijk@uu.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_12, © The Author(s) 2013

Using the method for the automatic identification of MWEs and the method for lexically representing MWEs, a database of MWEs of the Dutch language called *DuELME* has been constructed. It will be described in Sect. 12.5.

We end with concluding remarks in Sect. 12.6.

12.2 Multiword Expressions

Multi-word Expressions (MWEs) are word combinations with linguistic properties that cannot be predicted from the properties of the individual words or the way they have been combined. A word combination can, for example, have an unpredictable meaning (*de boeken neerleggen*, lit. 'to put down the books', meaning 'to declare oneself bankrupt'), it can have only limited usage (e.g. *met vriendelijke groet* 'kind regards', used as the closing of a letter), or it can have an unpredictable translation (*dikke darm* lit. 'thick intestine', 'large intestine'), etc.

MWEs do not necessarily consist of words that are adjacent, and the words making up an MWE need not always occur in the same order. This can be illustrated with the Dutch MWE *de boeken neerleggen* 'to declare oneself bankrupt'. This expression allows a canonical order with contiguous elements (as in (1a)), but it also allows other words to intervene between its components (as in (1b)), it allows permutations of its component words (as in (1c)), and combinations of permutations and intervention by other words that are not components of the MWE (as in (1d)):

(1) a. Saab heeft gisteren **de boeken neergelegd**
lit. 'Saab has yesterday the books down-laid'
b. Ik dacht dat Saab gisteren **de boeken** wilde **neerleggen**
lit. 'I thought that Saab yesterday the books wanted down-lay'
c. Saab **legde de boeken neer**
lit. 'Saab laid the books down'
d. Saab **legde** gisteren **de boeken neer**
lit. 'Saab laid yesterday the books down'

In addition, certain MWEs allow for (and require) controlled variation in lexical item choice, e.g. in expressions containing bound anaphora such as *zijn geduld verliezen* 'to lose one's temper', where the possessive pronoun varies depending on the subject (cf. *Ik verloor mijn/*jouw geduld, jij verloor *mijn/jouw geduld*, etc.), exactly as the English expression *to lose one's temper*.

Of course, not every MWE allows all of these options, and not all permutations of the components of an MWE are well-formed (e.g. one cannot have **Saab heeft neergelegd boeken de* lit.'Saab has down-laid books the').

One can account for such properties of MWEs by assigning an MWE the syntactic structure that it would have as a literal expression: it will then participate in the syntax as a normal expression, and permutations, intrusions by other words or phrases, etc. can occur just as they can occur with these expressions under

their literal interpretation.[1] Adopting this approach for the Dutch MWE *de boeken neerleggen* accounts immediately for the facts in (1) and for the ill-formedness of the example **Hij heeft neergelegd boeken de* given above, since this latter string is also ill-formed under the literal interpretation.

State-of-the art NLP systems do not deal adequately with expressions that are MWEs, and this forms a major obstacle for the successful application of NLP technologies. Reference [16] is titled: *Multiword expressions: a pain in the neck for NLP* and states that "Multiword expressions are a key problem for the development of large-scale, linguistically sound natural language processing technology".

Three problems must be solved to overcome this. First, an NLP system must have an implemented method of dealing with MWEs. This topic will not be dealt with in this paper. A lot of research has been spent on this, and for the purposes of this paper we simply observe that it has resulted in a wide variety of approaches in different grammatical frameworks and different implementations (see [13] for some relevant references).

Second, an NLP system does not 'know' which combinations of words form MWEs. Just providing a list of MWEs (an MWE lexicon) will not in general suffice because each domain has its own vocabulary and its own MWEs. There must thus be a way of dynamically creating MWE lexicons by identifying MWEs in new text corpora in a maximally automated way. The approach to this problem adopted here will be described in Sect. 12.3.

Third, each MWE identified must be represented lexically. The approach adopted here for this problem will be described in Sect. 12.4, and takes into account the fact that solutions to the first problem come in many different varieties in a wide range of grammatical frameworks and implementations.

12.3 Identification of MWEs and Their Properties

We need a method for identifying MWEs in a text corpus in a maximally automated manner. Since component words of an MWE can be inflected, we want to be able to identify multiple combinations of words as instances of the same MWE even if they contain differences with regard to inflection. Since MWEs can consist of nonadjacent words, and since the order in which the components of an MWE occur can vary, we want to be able to identify multiple combinations of words as instances of the same MWE even if the component words are nonadjacent or occur in different orders. Both of these requirements can be met if each sentence of the text corpus is assigned a syntactic structure and each occurrence of an inflected word form is assigned a lemma.

[1] There are, however, additional constraints on MWEs that do not hold for literal expressions. These will either follow from properties of the grammatical system if they involve general restrictions (e.g. they may follow from the fact that components of MWEs often have no independent meaning) or have to be stipulated individually for each MWE with idiosyncratic restrictions.

For experimenting with the identification methods we have used the Dutch CLEF corpus, a collection of newspaper articles from 1994 and 1995 taken from the Dutch daily newspapers *Algemeen Dagblad* and *NRC Handelsblad*. This corpus contains 80 million words and 4 million sentences. We have used the Alpino Parser[2] to automatically annotate each sentence of the corpus with a syntactic structure and each inflected word form token with a lemma.

The identification method takes as input a set of syntactic patterns and a fully parsed text corpus, and outputs a set of candidate MWEs in the form of tuples of lemmas together with a range of grammatical and statistical properties.

For example, for the syntactic pattern *NP_V* it will return tuples consisting of a word of syntactic category *verb* and a word of syntactic category *noun* that are likely candidates to form an MWE with the verb as the head and the noun as the head of the direct object noun phrase, together with statistics on occurring determiners and adjectives modifying the noun, etc.

Based on experiments with various machine learning techniques, it has been decided to apply a binary decision tree classifier to distinguish MWEs from non-MWEs [21].[3] The classifier characterises expressions as an MWE or as a non-MWE using a range of features that reflect or approximate properties of MWEs as reported in the linguistic literature. These include features of lexical affinity between MWE components, local context, morphosyntactic flexibility, and semantic compositionality. Lexical affinity between MWE components has been determined using *salience*, a variant of pointwise mutual information [10], and by a binary feature marking a small set of verbs as *support verbs* [8]. For *local context*, two measures proposed by Merlo and Leybold [12] to quantify *head dependence* are used, viz. the number of verbs that select a given complement, and the entropy of the distribution among the verbs that select for a given complement (cf. [21] for details). In addition, the relative frequency of the label most frequently assigned by the Alpino parser to the dependency relation between the head and the dependent is used. Since in Dutch PP complements are generally closer to the verb in verb-final context than PP adjuncts,[4] the relative frequency of the PP occurring adjacent to the verb group has also been taken into account.

Inflectional modifiability is quantified as follows, following [22]: the most characteristic realisation is simply the realisation of a phrase that occurs most frequently. The degree of modifiability is then expressed as the relative frequency of the most frequent realisation: a low relative frequency for the most frequent realisation indicates high modifiability, a high relative frequency indicates low modifiability.

Another feature used to determine morphosyntactic flexibility is the *passivisation* feature, which simply specifies the relative frequency of the occurrence of the

[2] See http://www.let.rug.nl/vannoord/alp/Alpino/ and [19].

[3] The classifier used is weka.classifiers.trees.j48 [23] which implements the C4.5 decision tree learning algorithm.

[4] [2, p. 107].

candidate expression as a passive. And finally, the pronominalisation feature records whether an NP has been realised as a pronoun.

For semantic compositionality two scores have been used as features, derived from work by Van De Cruys [17], who applied unsupervised clustering techniques to cluster nouns and verbs in semantically related classes. One score (semantic uniqueness) can be characterised as the ratio between the selectional preference of a selector for a selectee and the selectional preference of a selectee for its selector. The second score can be characterised as the selectional association between a selector and a selectee. See [18] for more details.

The data used for testing consist of the Dutch CLEF Corpus and the Twente News Corpus (TwNC[5]), which consists of 500 million word occurrences in newspaper and television news reports and which also has been fully parsed with the Alpino parser. These data have been annotated automatically using the existing lexical databases *Van Dale Lexical Information System* (VLIS)[6] and RBN [11]. The VLIS database contains more than 61,000 MWEs of various kinds (idioms, collocations, proverbs, etc.). From the RBN, app. 3,800 MWEs were extracted and used in the experiments. All expressions from VLIS and RBN have been parsed with the Alpino parser. From the resulting parse structures, sequences of tuples containing word form, lemma, PoS-label and position in the structure have been derived. In the test corpus, all expressions matching the input syntactic pattern have been identified. An absolute frequency threshold has been used to avoid noise introduced by very low frequency expressions. This threshold has to be determined empirically for each pattern as a function of the performance of the classifier.[7] For example, for the pattern *PP_V*, counting only types with frequency $\geq$10, the test corpus contains 4,969 types and for the pattern *NP_PP_V* it contains 3,519 types. Together this makes 8,488 types covering 1,140,800 tokens. If a candidate expression from the corpus matches with an entry from the set of tuple lists derived from the VLIS and RBN databases, it was marked as MWE, otherwise as non-MWE to serve as the gold standard.[8] Table 12.1 shows the distribution of MWEs and non-MWEs for two patterns.

As one can observe, when one uses a low frequency cut-off ($\geq$10), the proportion of non-MWEs equals 3/4 of the data, while with a higher frequency cut-off ($\geq$50), MWEs and non-MWEs occur equally often (as can be seen for the *NP_V* data in Table 12.1).

Experiments were carried out with different combinations of features. Semantic scores were initially left out. As a baseline, we use a classifier that always selects the

[5] http://wwwhome.cs.utwente.nl/~druid/TwNC/TwNC-main.html

[6] These data have been made available for this research by the Van Dale dictionary publishers. These data could only be used for research internal to the project.

[7] However, it is well known that many individual MWEs occur with very low frequency in a corpus. In fact, [21, p. 18] observes that this may double the number of MWEs. MWE identification methods should therefore also work for low frequency data. We leave this to further research.

[8] No distinction was made or could be made between the literal and the idiomatic uses of an expression. Therefore each expression that can be used as an MWE has been annotated as an MWE.

Table 12.1 Distribution of MWEs and non-MWEs for the patterns *(NP)_PP_V* and *NP_V*

Pattern	Freq	Types	MWEs	non-MWEs
(NP)_PP_V	≥10	8,488	1,910 (22.50 %)	6,578 (77.49 %)
NP_V	≥10	10,211	2,771 (27.13 %)	7,440 (72.86 %)
NP_V	≥50	1,769	917 (51.83 %)	852 (48.16 %)

Table 12.2 Major results for MWE identification for the pattern *(NP)_PP_V*

			MWE			Non-MWEs		
Features	Dataset	Acc	P	R	F	P	R	F
All	Test	82.07	0.62	0.47	0.54	0.86	0.91	0.89
All	All (10fcv)	82.99	0.67	0.48	0.56	0.86	0.93	0.89
All + semantic scores	Test	82.75	0.64	0.49	0.56	0.86	0.92	0.89
All + semantic scores	All (10fcv)	83.40	0.66	0.53	0.59	0.87	0.92	0.89
Baseline	All	77.49	0.00	0.00	0.00	1.00	1.00	1.00

most frequent class. The results specify *accuracy* (Acc), and for each class *precision* (P), *recall* (R) and *F-score* (F). It turned out that using *all* features yielded better results than using any of the tested subsets of features. With semantic scores added, accuracy increased a little more. Evaluation was carried out in two ways: In one set-up 60 % of the data was used for training and 40 % for testing. In a second set-up, all data were used for training and testing using ten-fold cross validation (10fcv). Table 12.2 lists the major results.[9]

The identification method operates on fully parsed sentences. It therefore can use sophisticated grammatical properties as features for the automatic identification of MWEs. The identification method yields a set of tuples that are characterised as MWEs, but it can also provide sophisticated grammatical and statistical properties for each MWE. This has in fact been done, using the CLEF corpus as well as the TwNC. For each candidate expression, a range of properties has been extracted that differs slightly with each pattern, but includes inter alia:[10]

1. The subcategorisation frame assigned by the Alpino parser to the head of the expression;
2. The absolute frequency of the tuple;
3. Corpus size
4. A list of heads of co-occurring subjects with frequencies;
5. For each complement:
   (a) Inflectional properties of the head of the complement, and their frequencies;
   (b) Diminutive information for the head of a nominal complement, and their frequencies;

[9]For more results and details, including an analysis of the relative contribution of the various features, we refer to [21].

[10]See [5, Appendix A] for a detailed description.

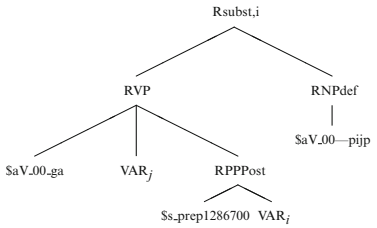


Fig. 12.1 Rosetta D-tree for the idiom *de pijp uitgaan* (simplified)

(c) Determiners co-occurring with the head of a nominal complement, and their frequencies;
(d) Heads of pre-modifiers of the head of the complement, and their frequencies; and
(e) Heads of post-modifiers of the head of the complement, and their frequencies.

The expressions identified as MWEs and their properties (9,451 were identified in the corpora) form the basis for the DuELME lexical database of MWEs, structured in accordance with the ECM.

12.4 Lexical Representation of MWEs

As we have seen above, assigning a syntactic structure to an MWE allows one to account for the fact that MWEs participate in syntax as normal expressions, i.e. allow for permutations, intrusions by other words and phrases, etc. The problem, however, with syntactic structures in NLP systems is that they are highly system specific. This has been shown in detail by Odijk [13] using the Rosetta machine translation system [15] as illustration. The Rosetta system requires, for idiomatic expressions, (1) a reference to a highly specific syntactic structure (cf. Fig. 12.1 for an example), and (2) a sequence of references to lexical entries of the lexicon of the system. In this sequence the presence/absence of these references, the order in the sequence, and the references themselves are all particular to the Rosetta system.

Lexical representations of MWEs that are highly specific to particular grammatical frameworks or concrete implementations are undesirable, since it requires effort in making such representations for each new NLP system again and again and the degree of reusability is low. No *de facto* standard for the lexical representation of MWEs currently exists. Various attempts have been made to develop a standard encoding for certain types of MWEs, especially within the ISLE[11] and XMELLT[12] projects. Reference [13] argues that these attempts are unlikely to be successful, because the structures assigned to the MWEs are highly theory-dependent and even

[11] www.ilc.cnr.it/EAGLES96/isle/

[12] www.cs.vassar.edu/~ide/XMELLT.html

within one grammatical framework, there will be many differences from implementation to implementation. Since most syntactic structures are fully specified tree structures, they are difficult to create and maintain. Reference [3] outlined an approach to represent MWEs in a form which can support precise HPSG, and which is also claimed to be reasonably transparent and reusable. Though their approach may work for certain types of MWEs, they fail to come up with a satisfying solution for representing MWEs.

The central idea behind the ECM is that a standardised representation does not prescribe the structure of an MWE, but backs off to a slightly weaker position, viz. it requires that it is specified which MWEs have the same syntactic structure. In short, it requires that equivalence classes of MWEs are created, based on whether they have the same syntactic structure. Having these equivalence classes reduces the problem of assigning a concrete structure and properties to an MWE to doing this for one instance of the class. And for this problem, the ECM includes a procedure that specifies how to derive that information to a large extent from the concrete system in which the MWE is incorporated.

The ECM thus specifies (1) a way to lexically represent MWEs, and (2) a procedure to incorporate MWEs into a concrete NLP system in a maximally automated manner.

An ECM-compatible lexical representation consists of

1. An MWE pattern, i.e. an identifier that uniquely identifies the structure of the MWE. The equivalence classes are defined with the help of these MWE patterns: MWEs with the same pattern belong to the same equivalence class;
2. A list of MWE components. This takes the form of a sequence of strings, each string representing the lexicon citation form of an MWE component. As to the order of the components, the proposal leaves the order free, but only imposes the requirement that the same order is used for each instance in the same equivalence class;
3. An example sentence that contains the MWE. The structure of the example sentence should be identical for each example sentence within the same equivalence class.

Next to the MWE description, we need a description of the MWE patterns. This is a list of MWE pattern descriptions, where each MWE pattern description consists of two parts:

1. An MWE pattern, and
2. Comments, i.e. free text, in which it is clarified why this MWE pattern is distinguished from others, further indications are given to avoid any possible ambiguities as to the nature of the MWE structure. It is even possible to supply a more or less formalised (partial) syntactic structure here, but the information in this field will be used by human beings and not be interpreted automatically.

This concludes the description of the lexical representation of an MWE in accordance with the ECM. Table 12.3 shows three instances of the same MWE equivalence class from Dutch, and gives a description of the MWE pattern used to define this equivalence class

Table 12.3 Three instances of MWE equivalence class *Pat1* and the description of the equivalence class

Pat.	Components	Example	Gloss	Translation
Pat1	de pijp uit gaan	Hij is de pijp uitgegaan	He is the pipe out-gone	'He died'
Pat1	het schip in gaan	Hij is het schip ingegaan	He is the ship in-gone	'He had bad luck'
Pat1	de boot in gaan	Hij is de boot ingegaan	He is the boat in-gone	'He had bad luck'

Pat.	Description
Pat1	Verb taking a subject and a directional adpositional phrase (PP). This PP is headed by a postposition and has as its complement a noun phrase consisting of a determiner and a singular noun.

The procedure to convert a class of ECM-compatible MWE descriptions into a class of MWE descriptions for a specific NLP-system consists of two parts: a manual part, and an automatic part. The manual part has to be carried out once for each MWE pattern, and requires human expertise of the language, of linguistics, and of the system into which the conversion is to be carried out. The automatic part has to be applied to all instances of each equivalence class.

The manual part of the conversion procedure for a given MWE pattern *P* consists of five steps:

1. Select an example sentence for MWE pattern *P*, and have it parsed by the system; select the right parse if there is more than one;
2. Define a transformation to turn the parse structure into the idiom structure;
3. Use the result of the parse to determine the unique identifiers of the lexical items used in the idiom;
4. Use the structure resulting from the parse to define a transformation to remove and/or reorder lexical items in the idiom component list;
5. Apply this transformation and make sure that the citation form of each lexical item equals the corresponding element on the transformed citation form list.

Observe that only one part of the first step of this procedure (*Select the right parse*) crucially requires human intervention.

The automatic part of the conversion procedure is applied to each instance of the equivalence class defined by idiom pattern *P*, and also consists of five steps:

1. Parse the example sentence of the idiom and check that it is identical to the parse tree for the example sentence used in the manual step, except for the lexical items;
2. Use the transformation defined above to turn the parse tree into the structure of the idiom;
3. Select the unique identifiers of the lexical items' base forms from the parse tree, in order;
4. Apply the idiom component transformation to the idiom component list;
5. Check that the citation form of each lexical item equals the corresponding element on the transformed idiom Component list.

The application of these procedures to real examples has been illustrated in detail in [13, 14]. However, the ECM as described above has one serious drawback, which can be solved by extending it with parameters into the *parameterised ECM*. A concrete example may help illustrate the problem and how the use of parameters resolves it. MWEs can contain nouns. In Dutch, nouns can be singular (*sg*) or plural (*pl*), and positive (*pos*) or diminutive (*dim*). In the ECM as described above a different equivalence class would be needed for each of these four cases (and even more if more than one noun occurs in a single MWE). By introducing two parameters for nouns (*sg/pl*, *pos/dim*), it is possible to group these four equivalence classes into a single equivalence superclass, and to have a single pattern for this superclass, which is parameterised for the properties of the noun (*sg/pl*; *pos/dim*). The extension with parameters introduces a little more theory and implementation specificity to the method, but it does so in a safe way: NLP systems that can make use of these parameters will profit from it, while systems that cannot make use of these parameters are not harmed since the original equivalence classes can still be identified. For the example given above the theory or implementation dependency that is introduced is that properties such as *sg/pl* and *pos/dim* on a noun are dealt with by rules applying to just the noun. It can be expected that many different grammatical frameworks share this assumption. The extension contributes to reducing the number of equivalence classes and increasing the number of members within equivalence classes. It will therefore reduce the number of MWEs that have to be dealt with manually and increase the number of MWEs that can be incorporated into an NLP system in a fully automatic manner. Reference [13] argued that the parameterised ECM is a feasible approach on the basis of data from English and a small set of data from Dutch, and the work reported on here for Dutch has confirmed this.

12.5 The DuELME Lexical Database

An ECM-compatible lexical database for Dutch MWEs has been created [6]. It is ECM-compatible because it classifies MWEs in equivalence classes based on their syntactic structure and uses lexical representations that cover all items required by the ECM. The database is corpus-based: the expressions included have been selected on the basis of their occurrence in the CLEF and TwNC corpora.

Six syntactic patterns frequently occurring in the parsed VLIS database have been selected as input patterns for the MWE identification algorithm. These patterns are defined in terms of dependency triples <head, dependent, dependency label> consisting of a *head* word and a *dependent* word, each of a specific syntactic category, and a *dependency label* to characterise the dependency relation between the words – cf. Table 12.4, where identical subscripts indicate that the same word must be involved and where *compl* is a variable for a range of labels that the dependency relation between a complement PP and a verb can have in Alpino syntactic structures.

Table 12.4 Input patterns

Pattern	Description
NP_V	<verb, noun, direct object>
(NP)_PP_V	<verb$_i$, adposition, *compl*> and optionally <verb$_i$, noun, direct object>
NP_NP_V	<verb$_i$, noun, indirect object> and <verb$_i$,noun, direct object>
A_N	<noun, adjective, modifier>
N_PP	<noun, adposition, modifier>
P_N_P	<adposition, noun$_i$, complement> and <noun$_i$, adposition, modifier>

Table 12.5 Number of candidate expressions and absolute frequency threshold by pattern

Pattern	Threshold	Count
NP_V	$f \geq 10$	3,894
(NP)_PP_V	$f \geq 10$	2,405
NP_NP_V	$f \geq 10$	202
A_N	$f \geq 50$	1,001
N_PP	$f \geq 30$	1,342
P_N_P	$f \geq 50$	607
Total		**9,461**

With these patterns as input, a wide variety of expression types can be extracted, since the patterns are underspecified for a lot of aspects, such as determiners, adjectival and adverbial modifiers, inflectional properties, etc. The resulting set of MWEs therefore requires many more MWE patterns than these six for an adequate description.

Using these patterns, candidate expressions in the form of tuples of head words and their properties have been identified in the corpora, in the way described in Sect. 12.3. The numbers of identified candidate expressions per pattern as well the absolute frequency thresholds used for each pattern are given in Table 12.5.

The 9,461 candidate expressions and their properties form the input to a process of manual selection of the expressions to include in the DuELME database, and of adapting candidate expressions. The criteria used in this manual selection process are criteria that follow the definition of MWE as given in Sect. 12.2: does the word combination have linguistic properties that cannot be predicted from the properties of the individual words or the way they have been combined? This manual step is necessary for many reasons. First, the identification method does not have 100 % accuracy, so the resulting list contains expressions that cannot be considered MWEs in accordance with the definition of MWEs as given in Sect. 12.2. Second, in many cases the expression as identified by the algorithm is incomplete, e.g., a determiner or modifying adjective that is obligatory for the MWE is not identified as part of the MWE by the identification method. The relevant information to further automate this is available (e.g. the properties contain statistics on the co-occurring determiners, modifiers, etc.), but it has not been used by the MWE identification

Table 12.6 MWEs containing *hand* and *hebben* as components

MWE	Gloss	Translation
zijn handen vol hebben aan	his hands full have on	'be fully occupied with'
de hand hebben in	the hand have in	'be responsible for'
de vrije hand hebben	the free hand have	'be free to do whatever one wants'
een gelukkige hand hebben	a lucky hand have	'be lucky'

Table 12.7 Coverage of ECs

Cov.(%)	#MWEs	#ECs	#parameterised ECs
50	2,616	101	10
60	3,139	166	16
70	3,662	272	25
80	4,186	441	38
85	4,447	572	48
90	4,709	785	63
95	4,970	1,046	87
100	5,232	1,308	140

algorithms.[13] In some cases, one output expression actually covers multiple different MWEs. An extreme case is the candidate expression characterised by the head of the direct object noun *hand* 'hand' and the head verb *hebben* 'have'. The examples from the corpus illustrate four different MWEs having these words as components, as illustrated in Table 12.6. Such examples cannot currently be distinguished as different MWEs by the identification method and have to be split manually into different entries.

The selection process resulted in 5,232 MWEs that have been included in the DuELME database. The selected and improved expressions have been analyzed for a classification by syntactic structure (equivalence classes, ECs). The parameterised ECM has been fully elaborated for Dutch. Eight parameters have been distinguished, each with multiple possible values, and in total 26 possible values.[14] The coverage of the ECs and the parameterised MWE classes is represented in Table 12.7.

In this table, we observe several things. The ECM without parameters requires a substantial amount of ECs to obtain a reasonable coverage, e.g. 785 to cover 90 % (or 4,709) of the lexicon entries. The amount of required ECs is a direct indicator for the amount of effort required to incorporate MWEs into an NLP system in accordance with the ECM procedure, and it is clear that without parameters this is too large to be realistically feasible. By the introduction of parameters, however,

[13] This is an area for future research, which is now possible since the relevant data are available.

[14] See [5, p. 36] for a complete overview.

the number of required ECs reduces dramatically, for 90 % coverage from 785 to 63. Of course, additional effort must be spent to deal with the parameters, but if all parameters can be optimally used, this just adds a fixed one-time effort of 26 operations (corresponding to the number of possible parameter values). This shows that the parameterised ECM approach is feasible, and reduces effort for incorporating MWEs into an NLP system considerably, confirming initial results in this direction presented by Odijk [13].

In the DuELME database, templates for syntactic structures have actually been added for each MWE pattern. These templates for syntactic structures are modeled after the syntactic dependency structures used initially in the CGN (Spoken Dutch Corpus, [7]) as well as in the D-Coi,[15] Lassy[16] and SoNaR[17] projects. These dependency structures have thus become a de facto standard for the representation of syntactic structures for Dutch. Adding such syntactic structures is not needed for the parameterised ECM, but they do no harm either. In fact, they can be beneficial for NLP systems that can deal with them. In particular, the first step of the manual part of the ECM incorporation method ('Select the right parse'), which is the only one requiring human intervention, can now become fully automatic, and thereby the whole manual part can become fully automated. In addition, for systems that use closely related syntactic structures, direct mappings can be defined.

Small experiments have been carried out to test incorporating MWEs into NLP systems. The experiments involved the Rosetta system and the Alpino system. For the Rosetta system this remained a paper exercise, since no running system could be made available. For Alpino, the incorporation worked effectively. It has also been tested, in a very small experiment which can at best be suggestive, how a system with incorporated MWEs performs in comparison to the system without these MWEs. This has been tested by measuring the *concept accuracy per sentence*(CA) as used in [20] for the Alpino system with the original Alpino lexicon and the Alpino system with an extended lexicon:

$$CA^i = 1 - \frac{D^i_f}{max(D^i_g, D^i_p)} \tag{12.1}$$

where D^i_p is the number of relations produced by the parser for sentence *i*, D^i_g is the number of relations in the treebank parse for sentence *i*, and D^i_f is the number of incorrect and missing relations produced by the parser for sentence *i*.

The results, summarised in Table 12.8, show that the concept accuracy of sentences containing an MWE increases significantly in a system with an extended lexicon, and the concept accuracy of sentences not containing MWEs does not decrease (in fact, also increases slightly) in a system with an extended lexicon.

[15] http://lands.let.ru.nl/projects/d-coi/

[16] http://www.let.rug.nl/vannoord/Lassy/

[17] http://lands.let.ru.nl/projects/SoNaR/

Table 12.8 Concept accuracy scores

Sample	Lexicon	CA(%)
MWEs	Alpino lexicon	82.85
	Extended lexicon	94.09
Non-MWEs	Alpino lexicon	95.83
	Extended lexicon	96.39

Such results are encouraging, but because of the small scale of the experiment, it should be confirmed by larger scale experiments before definitive conclusions can be drawn.

The DuELME database, a graphical user interface, and extensive documentation is available via the Dutch HLT Agency.[18] The database has been positively externally validated by CST, Copenhagen, i.e. been subjected to a check on formal and content aspects. In a CLARIN-NL[19] project, the database has been stored in a newly-developed XML representation that is compatible with the Lexical Markup Framework (LMF),[20] CMDI-compatible metadata[21] have been provided, and the data categories used in the database have been linked to data categories in the ISOCAT data category registry,[22] thus preparing its incorporation into the CLARIN research infrastructure and increasing the visibility, accessibility and the interoperability potential of the database. The graphical user interface has been adapted to work directly with this XML format. This version of DuELME can also be obtained via the Dutch HLT Agency.

12.6 Concluding Remarks

This paper has addressed problems that MWEs pose for NLP systems, more specifically the lack of large and rich formalised lexicons for multi-word expressions for use in NLP, and the lack of proper methods and tools to extend the lexicon of an NLP-system for multi-word expressions given a text corpus in a maximally automated manner. The paper has described innovative methods and tools for the automatic identification and lexical representation of multi-word expressions.

The identification methods operate on fully parsed sentences and can therefore use quite sophisticated manners of MWE identification that can abstract from different inflectional forms, differences in order, and deal with non-adjacent MWE components. Considerable progress has been achieved in this domain, and the

[18] http://www.tst-centrale.org/nl/producten/lexica/duelme/7-35

[19] http://www.clarin.nl

[20] http://www.lexicalmarkupframework.org/ and [4].

[21] CMDI stands for Components-based MetaData Infrastructure, cf. http://www.clarin.eu/cmdi and [1].

[22] http://www.isocat.org and [9].

methods developed have served as a basis for constructing a corpus-based lexical database for Dutch MWEs. However, there are still many opportunities for improvement. The most important topic for further research consists of finding methods for yielding more precise results for MWE identification, so that the manual step of selecting candidate MWEs can be significantly reduced.

The parameterised ECM has been investigated in detail on a large scale for Dutch. A full elaboration of the ECM parameters required for Dutch has been carried out. The incorporation method of the parameterised ECM has been tested in NLP systems, and an initial evaluation of the effect of MWEs incorporated in NLP systems has been carried out. Of course, there are opportunities for improvement here as well. It is in particular necessary to investigate in more detail how ECM parameters influence large scale integration of MWEs in NLP systems: to what extent can the parameters indeed be dealt with independently of the equivalence classes?

The paper describes a 5.000 entry corpus-based multi-word expression lexical database for Dutch developed using these methods. The database has been externally validated, and its usability has been evaluated in NLP-systems for Dutch. The MWE database developed fills a gap in existing lexical resources for Dutch. The generic methods and tools for MWE identification and lexical representation focus on Dutch, but they are largely language-independent and can also be used for other languages, new domains, and beyond this project. The research results and data described in this paper have therefore significantly contributed to strengthening the digital infrastructure for Dutch, and will continue to do so in the context of the CLARIN research infrastructure.

Acknowledgements The paper describes joint work by the IRME project team, and especially work carried out by Nicole Grégoire and Begoña Villada Moirón. I have liberally used material from reports, articles, PhDs etc. written by them, for which I am very grateful. I would also like to thank them and two anonymous reviewers for useful suggestions to improve this paper.

References

1. Broeder, D., Kemps-Snijders, M., Uytvanck, D.V., Windhouwer, M., Withers, P., Wittenburg, P., Zinn, C.: A data category registry- and component-based metadata framework. In: Calzolari, N., Maegaard, B., Mariani, J., Odijk, J., Choukri, K., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), Valletta, pp. 43–47. European Language Resources Association (ELRA), Valletta (2010)
2. Broekhuis, H.: Het voorzetselvoorwerp. Nederlandse Taalkunde **9**(2), 97–131 (2004)
3. Copestake, A., Lambeau, F., Villavicencio, A., Bond, F., Baldwin, T., Sag, I., Flickinger, D.: Multiword expressions: linguistic precision and reusability. In: Proceedings of the 3rd

International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, pp. 1941–7. ELRA (2002)
4. Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, M., Soria, C.: Lexical markup framework (LMF). In: Proceedings of LREC 2006, Genoa, pp. 233–236. ELRA, Genoa (2006)
5. Grégoire, N.: Untangling multiword expressions: a study on the representation and variation of Dutch multiword expressions. Phd, Utrecht University, Utrecht (2009). LOT Publication
6. Grégoire, N.: DuELME: A Dutch electronic lexicon of multiword expressions. J. Lang. Resour. Eval. **44**(1/2), 23–40 (2010). http://dx.doi.org/10.1007/s10579-009-9094-z
7. Hoekstra, H., Moortgat, M., Renmans, B., Schouppe, M., Schuurman, I., van der Wouden, T.: CGN syntactische annotatie. CGN report, Utrecht University, Utrecht (2003). http://lands.let.kun.nl/cgn/doc_Dutch/topics/version_1.0/annot/syntax/syn_prot.pdf
8. Hollebrandse, B.: Dutch light verb constructions. Master's thesis, Tilburg University, Tilburg (1993)
9. Kemps-Snijders, M., Windhouwer, M., Wright, S.: Principles of ISOcat, a data category registry (2010). Presentation at the RELISH Workshop Rendering Endangered Languages Lexicons Interoperable Through Standards Harmonization – Workshop on Lexicon Tools and Lexicon Standards, Nijmegen, 4–5 August 2010. http://www.mpi.nl/research/research-projects/language-archiving-technology/events/relish-workshop/program/ISOcat.pptx
10. Kilgarriff, A., Tugwell, D.: Word sketch: extraction & display of significant collocations for lexicography. In: Proceedings of the 39th ACL & 10th EACL workshop 'Collocation: Computational Extraction, Analysis and Exploitation', Toulouse, pp. 32–38. (2001)
11. Martin, W., Maks, I.: Referentie Bestand Nederlands: Documentatie. Report, TST Centrale (2005). http://www.tst-centrale.org/images/stories/producten/documentatie/rbn_documentatie_nl.pdf
12. Merlo, P., Leybold, M.: Automatic distinction of arguments and modifiers: the case of prepositional phrases. In: Proceedings of the Fifth Computational Natural Language Learning Workshop (CoNLL-2001), Toulouse, pp. 121–128 (2001)
13. Odijk, J.: A proposed standard for the lexical representation of idioms. In: Williams, G., Vessier, S. (eds.) EURALEX 2004 Proceedings, vol. I, pp. 153–164. Université de Bretagne Sud, Lorient (2004)
14. Odijk, J.: Reusable lexical representations for idioms. In: Lino, M.T., Xavier, M.F., Ferreira, F., Costa, R., Silva, R. (eds.) Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-2004), III, Lisbon, pp. 903–906. ELRA, Lisbon (2004)
15. Rosetta, M.: Compositional Translation, Kluwer International Series in Engineering and Computer Science (Natural Language Processing and Machine Translation), vol. 273. Kluwer, Dordrecht (1994)
16. Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: Multiword expressions: a pain in the neck for NLP. LinGO Working Paper 2001-03 (2001). http://lingo.stanford.edu/csli/pubs/WP-2001-03.ps.gz
17. Van De Cruys, T.: Semantic clustering in Dutch. In: Sima'an, K., de Rijke, M., Scha, R., van Son, R. (eds.) Proceedings of the Sixteenth Computational Linguistics in the Netherlands (CLIN), pp. 17–32. University of Amsterdam, Amsterdam (2006)
18. Van de Cruys, T., Villada Moirón, B.: Semantics-based multiword expression extraction. In: Grégoire, N., Evert, S., Kim, S. (eds.) Proceedings of the Workshop 'A Broader Perspective on Multiword Expressions', Prague, pp. 25–32. ACL, Prague (2007)
19. van der Beek, L., Bouma, G., van Noord, G.: Een brede computationele grammatica voor het Nederlands. Nederlandse Taalkunde **7**, 353–374 (2002)
20. van Noord, G.: At last parsing is now operational. In: Mertens, P., Fairon, C., Dister, A., Watrin, P. (eds.) TALN06 Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles, Leuven, pp. 20–42 (2006)
21. Villada Moirón, B.: Evaluation of a machine-learning algorithm for MWE identification. Decision trees. STEVIN-IRME Deliverable 1.3, Alfa-Informatica, Groningen (2006). http://www-uilots.let.uu.nl/irme/documentation/Deliverables/BVM_D1-3.pdf

22. Wermter, J., Hahn, U.: Collocation extraction based on modifiability statistics. In: Proceedings of COLING 2004, Geneva (2004)
23. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Chapter 13
The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch

Nelleke Oostdijk, Martin Reynaert, Véronique Hoste, and Ineke Schuurman

13.1 Introduction

Around the turn of the century the Dutch language Union commissioned a survey that aimed to take stock of the availability of basic language resources for the Dutch language. Daelemans and Strik [5] found that Dutch, compared to other languages, was lagging behind. While the Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN; [25]) addressed the need for spoken language data, the dire need for a large corpus of written Dutch persisted and the construction of a multi-purpose reference corpus tailored to the needs of the scientific research as well as commercial development communities was identified as a top priority in the creation of an infrastructure for R&D in Dutch HLT.

The reference corpus, it was envisaged, should be a well-structured, balanced collection of texts tailored to the uses to which the corpus is going to be put. The contents of the corpus as well as the nature of the annotations to be provided were to be largely determined by the needs of ongoing and projected research and development in the fields of corpus-based natural language processing. Applications such as information extraction, question-answering, document classification, and

N. Oostdijk (✉)
Radboud University Nijmegen, Nijmegen, The Netherlands
e-mail: n.oostdijk@let.ru.nl

M. Reynaert
Tilburg University, Tilburg, The Netherlands
e-mail: reynaert@uvt.nl

V. Hoste
University College Ghent and Ghent University, Ghent, Belgium
e-mail: veronique.hoste@hogent.be

I. Schuurman
KU Leuven, Leuven, Belgium
e-mail: ineke.schuurman@ccl.kuleuven.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_13,

automatic abstracting that are based on underlying corpus-based techniques were expected to benefit from the large-scale analysis of particular features in the corpus. Apart from supporting corpus-based modeling, the corpus was to constitute a test bed for evaluating applications, whether or not these applications are corpus-based.

On the surface, all stakeholders agree that a large reference corpus of written Dutch would be invaluable for linguistic research and the development of profitable services that require advanced language technology. However, as soon as one starts making preparations for the collection of the text, and the definition of the minimal set of meta-data and annotation layers, it appears that different purposes may very well translate into very different requirements. A very large, balanced, richly annotated multi-purpose reference corpus is very different from the task-specific corpora that have been built in – for example – the DARPA programmes and the European CLEF programme. What is more, while some of the stakeholders (e.g. linguists, application developers and system integrators) may be able to formulate requirements and desires in the terms of their own disciplines and business areas, it is not straightforward to translate these formulations into technical requirements for a reference corpus. This is one of the reasons why in 2005 the STEVIN Dutch Language Corpus Initiative (D-Coi) project was initiated.

Although there were as yet no examples of the type of reference corpus aimed at, it was, of course, possible to derive boundary conditions from experiences with existing corpora and the major trends in the development of linguistics and language technology.[1] Thus, a modern reference corpus should not only sample texts from conventional media such as books and newspapers, but also from electronic media, such as web pages, chat boxes, email, etc. It was evident that inclusion of texts from these sources would pose (new) problems related to IPR, and that they would require the development of novel tools for the detection and annotation of typos, non-words, and similar phenomena that are less prominent in well-edited texts from the conventional printed media.

The D-Coi project was a pilot project that aimed to produce a blueprint for the construction of a 500-million-word (500 MW) reference corpus of written Dutch. This entailed the design of the corpus and the development (or adaptation) of protocols, procedures and tools that are needed for sampling data, cleaning up, converting file formats, marking up, annotating, post-editing, and validating the data.[2] In order to support these developments a 50 MW pilot corpus was compiled, parts of which were enriched with linguistic annotations. The pilot corpus should demonstrate the feasibility of the approach. It provided the necessary testing

[1] At the time (i.e. in 2004, at the start of the STEVIN programme) the American National Corpus (ANC; [16]) was probably closest to what was envisaged for the Dutch reference corpus as it also intended to include data from electronic media.

[2] Already in the planning phase, we realised the importance of adhering to (inter)national standards and best practices. Subsequently, wherever possible we have tried to relate to and build upon (the results of) other projects as well as re-use of resources and tools. Especially the CGN project has been particularly influential.

ground on the basis of which feedback could be obtained about the adequacy and practicability of the procedures for acquiring material and handling IPR, as well as of various annotation schemes and procedures, and the level of success with which tools can be applied. Moreover, it served to establish the usefulness of this type of resource and annotations for different types of HLT research and the development of applications.

There can be no doubt that as preparatory project the D-Coi project has been very useful. It provided the opportunity to come up with a design for a reference corpus in close consultation with the user community. Moreover, the compilation of the pilot corpus gave us hands-on experience with the work ahead of us, some facets of which we had underestimated before. With the insights gained we got a better view of what realistically could be done and what not. This has definitely proven to be advantageous as we were much better prepared when in 2008 we undertook the actual construction of the full reference corpus in the SoNaR project.[3]

In what follows we describe the various phases in the construction of the reference corpus. In Sect. 13.2 different aspects related to corpus design and data acquisition are discussed. Section 13.3 focuses on corpus (pre)processing, paying attention to the steps taken to handle various text formats and arrive at a standard XML version. Section 13.4 describes the various types of annotation and how they came about. Finally, Sect. 13.5 concludes this chapter.

13.2 Corpus Design and Data Acquisition

In this section we describe the design of the written Dutch reference corpus and its implementation, relating the strategies adopted in collecting different text types (including a wide range of texts from both traditional and new media) and the experiences in the acquisition and arrangement of IPR.

13.2.1 Corpus Design

The Dutch reference corpus was intended to serve as a general reference for studies involving language and language use. The corpus should provide a balanced account of the standard language and the variation that occurs within it. In doing so, it allows researchers investigating language use in a particular domain (e.g. medicine) or register (e.g. academic writing), or by a specific group (e.g. professional translators)

[3]The acronym SoNaR stands for STEVIN Nederlandstalig Referentiecorpus, i.e. STEVIN Dutch Reference Corpus.

to relate their data and findings to the general reference corpus. The corpus was also intended to play a role in the benchmarking of tools and annotations.[4]

The design of the Dutch reference corpus profited from the experiences in other large scale projects directed at the compilation of corpora (e.g. the British National Corpus, BNC – [1], the ANC and the CGN). In addition, consultation of the user community contributed to establishing needs and priorities.

The user requirements study [28] constituted a crucial step in the process of designing a Dutch reference corpus. The inventory of the needs and desires of linguists and members of the Dutch HLT community made by means of a web questionnaire, followed by consultation of the different user communities in focus groups, helped us decide on the priorities that should be set. Through the involvement of (potential) future users in this early stage we expected to avoid oversights and shortcomings that could easily result from too narrow a view on design issues and a limited awareness of existing needs. Equally important, user involvement throughout the design stages of corpus creation would contribute to generate the necessary support for such an undertaking and knowledge transfer.

The design was ambitious as it aimed at a 500 MW reference corpus of contemporary standard written Dutch as encountered in texts (i.e. stretches of running discourse) originating from the Dutch speaking language area in Flanders and the Netherlands as well as Dutch translations published in and targeted at this area. Texts were to be included from more conventional genres and text types as well as from the new media. The corpus was to include native speaker language and the language of (professional) translators. It was intended that approximately two-thirds of the texts would originate from the Netherlands and one-third from Flanders. Only texts were to be included that had appeared from the year 1954 onwards.[5]

The design envisaged the inclusion of texts written to be read as well as texts written to be spoken, published and unpublished texts, and also of texts that had appeared in print or in electronic form, or had been typed (cf. Table 13.1). As we aimed for a balanced, multi-purpose corpus, the corpus was to include a wide range of text types, from books, magazines and periodicals to brochures, manuals and theses, and from websites and press releases to SMS messages and chats. Moreover, the sheer size of the corpus made it possible to aim for the inclusion of full texts rather than text samples, leaving it to future users of the corpus to decide whether to use a text in its entirety or to use only a select part of it that meets the sampling criteria that follow more directly from a specific research question.

In the specification of the design of the Dutch reference corpus we intentionally deviated from other previous corpus designs for reference corpora such as the BNC

[4]Cf. the definition of a reference corpus provided by EAGLES: "*A reference corpus is one that is designed to provide comprehensive information about a language. It aims to be large enough to represent all the relevant varieties of the language, and the characteristic vocabulary, so that it can be used as a basis for reliable grammars, dictionaries, thesauri and other language reference materials.*"

[5]In the year 1954 a major spelling reform was put into effect, as a result of which from this year onwards a common spelling of the Dutch language came into use in Belgium and the Netherlands.

Table 13.1 Overall corpus design in terms of three main design criteria, viz. intended delivery of the texts included, whether they were published or not, and the primary mode (electronic, printed or typed)

Written to be read	Published	Electronic
492.5 MW	362.5 MW	177.5 MW
		Printed
		185.0 MW
	Unpublished	Electronic
	130.0 MW	100.0 MW
		Printed
		10.0 MW
		Typed
		20.0 MW
Written to be spoken	Unpublished	Electronic
7.5 MW	7.5 MW	2.5 MW
		Typed
		5.0 MW

and ANC. Especially the inclusion of much larger volumes of electronic texts, both published and unpublished, caused experts from the Center for Sprogteknology (CST, Copenhagen) charged with the evaluation of the design to raise questions as to its justification. Concerns were voiced as regards the effect the inclusion of such high quantities of electronic text would have on corpus quality, the arrangement of IPR, and thus on the representativeness and the balance of the corpus. At the same time the experts were receptive to the idea of an alternative design as they could well imagine that

> "the corpus will be contributing to, or may even be setting future best practices with regard to the proportional representation of electronic publications in reference corpora, because the existing guidelines that can be derived from the current large reference corpora, BNC and ANC, may need some additions. Text types like e-mail and discussion lists, chat and SMS are highly influenced by the intentions of quick, personal communication and by the requirements/limitations of the medium of communication as regards their functional style and language which differentiate them from traditional written text types. However, the need for novel NLP tools appropriate for new communication channels such as web chats, blogs, etc. justifies the high inclusion rate of such text types in a corpus intended to serve as a linguistic resource for the development of such NLP methods and tools." [2, p. 7]

In the course of the SoNaR project the corpus design originally conceived in the D-Coi project was modified.[6] There were several reasons for this. As we found that preprocessing typed texts was very laborious, time-consuming and error-prone, we decided to refrain from including this type of material. In other cases, such as

[6] An overview of the original design can be found in Table A.1 in the Appendix. For a detailed description and motivation we refer to [27].

with SMS messages where we found that the acquisition was quite problematic we decided on more realistic targets (e.g. 50,000 SMS texts instead of 5 MW).[7] Finally, the enormous flight Twitter has taken was a development we did not anticipate and was cause for modifying the design. In fact, the original design did not envisage the collection of tweets at all.

13.2.2 IPR

The reference corpus is intended to serve and be available to the wider research community. Therefore, considerable efforts were put into the settlement of the intellectual property rights (IPR). This was done in close collaboration with the Dutch HLT Agency who is responsible for the distribution of the corpus and its future maintenance. While the HLT Agency arranges the licences with prospective end users (academics and other non-profit institutes but also commercial parties) before granting them access to the data, it was the responsibility of the corpus compilers to make sure that IPR was settled with the content owners who agreed to have their texts included in the corpus.[8] To this end, the HLT Agency provided model contracts that the corpus compilers could use.

IPR had to be arranged for texts from all kinds of sources, both in the public but also in the more private domain. With texts from the conventional printed media (such as books, magazines, newspapers) the situation as regards IPR is fairly clear.[9] IPR can usually be settled through the publisher. For texts that are born-digital and are apparently freely available on the internet (such as websites and discussion fora) arranging IPR, we found, is rather more tricky. In some cases IPR lies with the site owner as contributors at some point have consented to have their rights carried over. However, in many such cases it is unclear whether the data may be passed on to a third party. In other cases no apparent IPR arrangements have been made. As a result the IPR status of these data remains unclear and the rights probably remain with the original authors/contributors. With data from for example chat and SMS individual people must give their consent. It is especially with these more private types of data that people were hesitant to have their texts included in a corpus. Anonymisation of the data was considered but not further pursued as this would involve a great deal of work, while it would seriously impact on the authenticity of the data.

[7] cf. Sect. 13.2.3 and 13.3.1.

[8] It should be noted that on principle we never paid for the acquisition of data and the settlement of IPR. Sometimes we would pay a small fee for the extra work that a text provider put into delivering the texts in a form that for us was easier to handle. In the SMS campaign there was the chance of a prize for those who contributed data.

[9] Although things may be complicated when texts have been digitised and placed on the internet (as for example those in DBNL – Digitale Bibliotheek Nederlandse Letteren, http://www.dbnl.org/).

In a number of cases there was no need to follow up on IPR matters as the texts were already available under some kind of licence, such as GNU GPL or Creative Commons, or by arrangement of law (the public's right to information).

13.2.3 Acquisition

Data acquisition has proven to be quite a formidable task. Ideally acquisition would be directed at texts that are already available in an open digital format so that the amount of work that must be put into making the text accessible can be reduced to a minimum. In actual practice we found that if we were to restrict the selection of data in this fashion this would seriously affect the balancedness of the corpus, especially since even with major publishers today the bulk of their holdings are not in (an open) digital format. In the acquisition process the primary aim was to identify and acquire texts that would fit the corpus design. And although we maintained a preference for formats that were readily accessible, we did not shy away from texts in formats that we knew would require considerable effort to preprocess.

As we wanted the corpus to reflect the large degree of variation found not only between text types but also within one and the same text type, acquisition efforts were directed at including texts from a large variety of sources. The identification of potential text providers was done on an ad hoc basis using various means available to us. Thus the networks of project members and associates were tapped into, contacts were established and major agreements arranged with television broadcasting companies, the conglomerate of national newspapers, major publishers of periodicals and other large text providers, while many other candidates were identified on the basis of their web presence. As a result of the attention the creation of the reference corpus attracted from the media, occasionally we would be approached by people offering data or giving pointers to interesting data sets. Where we were aware of other text collections that held Dutch data representative of specific text types (such as JRC-Acquis for legal texts or the OPUS Corpus which includes Dutch subtitles), we have pursued the inclusion of these data.[10] This course of action was motivated by the idea that in the SoNaR project we would impact an added value in yielding the XML uniform to the other data in the reference corpus, but also through the tokenisation and further linguistic annotations we provide.

For successful acquisition we found there is no single standard recipe. Different types of text and text providers require different approaches. Moreover, there are cultural differences: where potential text providers in Flanders may be persuaded to donate their texts arguing that the future of the Dutch language is under threat, in the

[10]JRC-Acquis is a collection of parallel texts from the EU comprising "the contents, principles and political objectives of the Treaties; EU legislation; declarations and resolutions; international agreements; acts and common objectives" [44]. The OPUS Corpus is an open parallel corpus which is publicly available. See also http://opus.lingfil.uu.se/.

Netherlands the fact that by donating texts a contribution is made to science is what is found particularly appealing. The strategies used and experiences gained in the SoNaR project in approaching potential text providers, negotiating and successfully settling IPR have been documented in [8].[11]

Of course at some point arrangements must be made for the actual transfer of the acquired data. What is all too readily overlooked is that the ease with which data can be transferred from the text provider to the corpus compiler can be a decisive factor in the successful acquisition of texts. If transfer is complex and requires that effort be put into it on the part of the text provider, chances are that the provider will refrain from doing so.

There are various ways of making the transfer of data easy for data providers. One example is the use of a drop box. Although the SoNaR drop box we had at our disposal was introduced rather late in the project it has demonstrated its usefulness.[12] It provided an easy interface to the text provider for uploading the (archives of) text files and for providing, at his/her own discretion some personal information for inclusion in the metadata. After submission, the text provider received a thank-you email which further contained the actual text of the IPR-agreement the text was subject to. Another example of how the transfer of data may be made easy is the way in which by means of an existing application SMS texts could be uploaded directly from Android mobile phones onto the SoNaR website.[13]

At the beginning of this section it was observed that data acquisition was a formidable task. Indeed, identifying and acquiring the necessary data and arranging IPR for a corpus of 500 million words represents a major challenge. Yet, as such it is not so much the large quantity of data that one should be in awe of, it is the quantity combined with the diversity of text types that the corpus comprises that is truly ambitious. All through the project the balancedness of the corpus has been a concern. Especially with texts directly obtained from the internet the amount of data tended to rapidly exceed the quantity envisaged in the corpus design. For example, the largest Flemish internet forum that we managed to arrange IPR with, by itself holds well over 500 million words of text. On the other hand, other text types were really hard to come by and were constantly at risk of being struck off the acquisition list. The corpus design was therefore used to control for balancedness and to ensure that apart from quantity there would be sufficient diversity: in a number of cases (such as the Flemish internet forum) only a fraction of the material is actual part of the 500 MW SoNaR corpus; the rest of the data is regarded as surplus. To the extent

[11] For the acquisition of tweets and SMS, special campaigns were organised (see [35, 47]).

[12] URL: http://webservices.ticc.uvt.nl/sonar/

[13] The original application was developed by the National University of Singapore. It was adapted for use in the SoNaR project. Adaptation consisted primarily in translating the operating instructions for uploading SMS texts. Linked to this is a SoNaR website on which more information about the project and more instructions specific to different kinds of mobile (smart)phones could be found (URL: http://www.sonarproject.nl/).

possible within the limitations of the project these data have been processed in the same manner and are available to those for whom there is never enough data.

Apart from having the data in the corpus represent various text types and topic domains, we also wanted the corpus to include both data originating from Flanders and data from the Netherlands. In a number of cases, as for example with the data from Wikipedia or JRC-Acquis, it was impossible to establish the origin.

All the text data files that were collected were gathered centrally and stored along with available metadata (such as content provider, date downloaded, original filename). An overview of the composition of the reference corpus can be found in Table A.1 in the Appendix.

13.2.4 Pilot Corpus

For the pilot corpus no separate design was made. In fact, the compilation of the pilot corpus ran very much in parallel to the work done in relation to the design of the 500 MW corpus and the development of procedures and the drafting of contracts that could be used for settling IPR matters. Given the primary aim of the pilot corpus, the main concern was that the corpus should be varied enough to be able to test the various procedures and protocols so as to avoid any omissions or oversights that might affect the compilation of the reference corpus.

In the compilation of the D-Coi pilot corpus, we found that IPR issues frustrated the acquisition process. In order to make sure that sufficient material would be available we therefore resorted to a more opportunistic approach of acquiring data. This involved focusing on data that were already available in the public domain (e.g. under a GPL or Creative Commons licence) or considered low-risk, such as texts found on public websites maintained by the government and public services.[14] Some genres and text types, however, remain underrepresented in the pilot corpus or do not occur in it at all. The latter is true for example for chat, email and SMS. Moreover, the corpus comprises relatively few Flemish data. An overview of the composition of the pilot corpus can be found in Table A.1 in the Appendix. The pilot corpus is described in more detail in [26].

13.3 Corpus (Pre)Processing

In this section we describe the various steps in the preprocessing of the corpus, from the stage where texts have been acquired and delivered in their original formats, up to the point where they are available in a uniform XML format.

[14] 'Low-risk' meaning that remaining IPR issues could be expected to be resolved in the not too distant future.

13.3.1 Text Conversion

The first step to be taken once the data had been acquired was to make the incoming data stream suitable for further upstream processing. It involved the conversion from the different file formats encountered such as PDF, MS-Word, HTML and XML to a uniform XML format.[15] This uniform format should allow us to store metadata and the text itself along with linguistic annotations from later processing stages. Moreover, it provided the means to perform XML validation after each processing stage: first after the conversion from original file format to the target format, and then again whenever new annotations had been added. Especially the validation after the first conversion appeared to be a crucial one in order to prevent that the processing chain was jammed due to incorrect conversions.

Putting much effort in the development of conversion tools was regarded outside the scope of the project. However, the conversion from original format to target XML appeared to be rather problematic in a substantial number of cases. Given the data quantities aimed at, an approach that uses a (semi-)manual format conversion procedure was not regarded a realistic option. Therefore the approach was to use existing conversion tools and repair conversion damage wherever possible. For a large proportion of the data this procedure worked quite well. Sometimes only minor adaptations to the post-processing tools were required in order to fix a validation problem for many files. Some parts of the collected data, however, had to be temporarily marked as unsuitable for further processing as it would take too much time to adapt the post-processing tools. Especially the conversion of the PDF formatted files appeared to be problematic. Publicly available tools such as pdf2html that allow for the conversion from PDF to some other format often have problems with columns, line-breaks, and headers and footers, producing output that is very hard to repair. On the other hand, as moving away from abundantly available content in PDF format would seriously limit the possibilities in finding a balance over text data types, the approach was to do PDF conversion semi-automatically for a small part of the collection. A varying amount of effort was required to convert other formats successfully to the target file format.

Progress of the work could be monitored by all project partners via a simple PHP web-interface[16] on a MYSQL database containing the relevant information for each file such as the raw word counts, validation status for each level, and total word counts (grand total, counts per document group, validated, etc.). The database was synchronised with the information in the D-Coi/SoNaR file system so that project partners could immediately fetch data that became available for their processing stage. The database and web-interface served as intermediate documentation of the work done.

[15] In the D-Coi project the XML format previously used in the CGN project was adopted with some slight changes. In SoNaR the D-Coi XML format was again modified (cf. also Sect. 13.5).

[16] URL: http://hmi.ewi.utwente.nl/searchd-coi

13.3.2 Text Tokenisation and Sentence Splitting

A major aim of the first conversion step to XML was to have titles and paragraphs identified as such. This is because most tokenisers, our own included, may fail to properly recognise titles and because the sentence splitting process expects a paragraph to consist of at least one full sentence. Failure in the first conversion step to recognise that a paragraph in TXT format is split up into n lines by newline characters, results in n XML paragraphs being defined. This is unrecoverable to the tokeniser. This fact can mostly be detected by the ratio of sentences identified after tokenisation in comparison to the number of paragraphs in the non-tokenised version. In such cases both unsuccessful versions were discarded and new ones produced semi-automatically by means of minimal, manual pre-annotation of the raw TXT version of the documents.

The rule-based tokeniser used was developed at the Induction of Linguistic Knowledge research team at Tilburg University prior to the D-Coi project. It was slightly adapted to the needs of the D-Coi/SoNaR projects on the basis of evaluations conducted by means of TOKEVAL, a tokeniser evaluator developed during the project in order to evaluate the available sentence splitters and tokenisers.[17] A very good alternative to the ILK tokeniser (ILKTOK), is the tokeniser that is available in the Alpino Parser distribution. As neither of the sentence-splitters/tokenisers available to us handled XML, we developed a wrapper program (WRAPDCOITOK) that deals with the incoming XML stream, sends the actual text to the sentence splitter/tokeniser, receives the outcoming sentences and tokens and wraps them in the appropriate XML. This scheme further allows for collecting sentence and word type statistics and for word type normalisation during the tokenisation step.

13.3.3 Text Normalisation and Correction

During the D-Coi project we developed CICCL, which is a set of programs for identifying various types of primarily typographical errors in a large corpus. CICCL stands for 'Corpus-Induced Corpus Clean-up' and has in part been described in [32]. Assumptions underlying this work are: (1) that no resources other than corpus-derived n-gram lists are available, (2) that the task can be performed on the basis of these resources only, to a satisfactory degree, (3) that in order to show that this is so, one needs to measure not only the system's accuracy in retrieving non-word variations for any given valid word in the language, but also its capabilities of distinguishing between what is most likely a valid word and what is not.

[17] These and other tools developed in the D-Coi project are available from http://ilk.uvt.nl, asareourtechnicalreports.

Where diacritics are missing and the word form without diacritics is not a valid word in its own right, fully automatic replacement was mostly possible and has been effected. This was performed for the words requiring diacritics which are listed in the [57], i.e. the official 'Word list of the Dutch Language'. Also we have a list of about 16,500 known typos for Dutch and most of the selections have been screened for these.

In the SoNaR project, text correction was performed more thoroughly, i.e. all divergent spelling variants were automatically lined up with their canonical form by means of TICCL (Text-Induced Corpus Clean-up), which was introduced in [33]. In the course of the project we have continued to develop new approaches to large scale corpus clean-up on the lexical level. In [34] we report on a new approach to spelling correction which focuses not on finding possible spelling variants for one particular word, but rather on extracting all the word pairs from a corpus that display a particular difference in the bag of characters making up the words in the pairs. This is done exhaustively for all the possible character differences given a particular target edit distance, e.g. an edit distance of 2 edits means that there are about 120K possible differences or what we call character confusions to be examined.

13.3.4 Language Recognition

Where deemed necessary or desirable during processing, we have applied the TextCat tool for language recognition.[18] Depending on the source and origin of the texts this was variously applied at document or paragraph level. Language recognition was never applied at sub-sentential level. However, in the Wikipedia texts, paragraphs containing foreign UTF-8 characters above a certain threshold were summarily removed, not on the basis of a TextCat classification but on encoding alone.

For some batches, notably the posts from a Flemish internet forum primarily dedicated to popular music and thus mainly to adolescents, TextCat was used to classify all posts separately. We found that over half received the following TextCat verdict: "I do not know this language". The language in question almost infallibly being a dialectical variety of the poster's specific internet idiolect. These posts were included and their TextCat categorisation was included in the metadata.

13.4 Corpus Annotation

This section describes the various types of annotations that were added to either the full reference corpus (the SoNaR-500 corpus for short), or one of two subsets: the D-Coi pilot corpus or a set of one million words (the SoNaR-1 corpus for short, cf.

[18] TextCat is available from http://www.let.rug.nl/vannoord/TextCat/

Table 13.2 Composition of the SoNaR-1 corpus. In all SoNaR-1 comprises 1,000,437 words

Text type	# words	Text type	# words
Administrative texts	28,951	Manuals	5,698
Autocues	184,880	Newsletters	5,808
Brochures	67,095	Newspapers	37,241
E-magazines and e-newsletters	12,769	Policy documents	30,021
External communication	56,287	Press releases	15,015
Instructive texts	28,871	Proceedings	6,982
Journalistic texts	81,682	Reports	20,662
Legal texts	6,468	Websites	32,222
Magazines	117,244	Wikipedia	260,533

Table 13.2). A decisive factor as regards what annotations were added to which dataset was the availability of tools that were sufficiently mature to allow large scale, fully automatic annotation. For part of speech tagging and lemmatisation, and named entity recognition this is (now) the case. For syntactic and semantic annotation, however, the annotation process is at best semi-automatic (that is, when aiming for annotations of high quality).

Since it is generally believed that the lack of a syntactically and semantically annotated corpus of reasonable size (min. 1 MW) is a major impediment for the development of academic and commercial tools for natural language processing applied to the Dutch language, we invested in these types of annotations. The SoNaR-1 corpus was syntactically annotated and manually verified in the Lassy project while in the SoNaR project four semantic annotation layers were added. These layers, which include the annotation of named entities, co-referential relations, semantic roles and spatio-temporal relations, were completely manually checked. Where tools were available for pre-annotation, the task was redefined as a correction task.

13.4.1 Part-of-Speech Tagging and Lemmatisation

For the tagging and lemmatisation of the reference corpus we aimed to yield annotations that were compatible to those in the CGN project. To the extent possible we wanted to re-use the tag set as well as the annotation tools and protocols for the human annotators. The tag set used to tag the reference corpus is essentially the same as that used for the Spoken Dutch Corpus (CGN), be it that a few tags were added to handle phenomena that do not occur in spoken language such as abbreviations and symbols [50]. Moreover, some tags that already existed in the original CGN tag set in the D-Coi/SoNaR version cover additional phenomena.

In the D-Coi project the CGN tagger/lemmatiser was adapted and retrained so that it would be able to cope with written text. This new version of the tagger/lemmatiser, which went by the name of Tadpole, was used to tag and

lemmatise the entire D-Coi pilot corpus.[19] PoS tagging with Tadpole reached an accuracy of 96.5 % correct tags (98.6 % correct on main tag) on unseen text.

For part of the pilot corpus (500,000 words) the tagging output of Tadpole was manually verified.[20] This was done with the idea that it would provide us with a qualitative analysis of its strengths and weaknesses, something we thought was of particular importance since the tagging-lemmatisation of the reference corpus would be done fully automatically (the sheer size of the corpus prohibited manual verification).

The task of manually verifying the tags was a bit of a challenge: the high accuracy output attained by Tadpole made it hard to find the few mistakes left, especially when looking through the tags one by one. We therefore deployed a tool that focused on suspect tags only (identified by a low confidence value).

The output of the tagger consisted of PoS tagged files, containing all possible tags for each token, together with the probability of that tag. We developed a tool for the manual correction of these automatically generated PoS tagged files. This tool takes a PoS tagged file as input, together with a threshold value. It presents the human annotator only with those cases where more than one possible tag has an above-threshold probability. All other cases where more than one tag is generated by the tagger, or those cases where only one tag is generated, are not presented to the annotator, resulting in a markedly lower workload.

We performed a small experiment to determine at which value we best set the threshold: a threshold value of 0.06 results in a reduction of the number of decisions to be made by the human annotator with 28 %, while skipping a mere 1 % of errors which are not presented to the annotator. This shows that, with the benefit of a tagger well-trained on a large volume of manually checked training material, we can manually check much larger amounts of data in the same time, missing hardly any errors. While following this procedure, all manually corrected material is regularly checked against a blacklist of typical errors made by the tagger, particularly on multi-word named entities and high-frequency ambiguous function words such as *dat* ('that', having the same ambiguity as in English) which the tagger sometimes tags incorrectly but with high confidence.

Except for some types of data originating from the new media, the reference corpus was tagged and lemmatised automatically using Tadpole's successor FROG.[21] In view of the huge amount of data and the high quality of FROG's output we refrained from any manual verification of the tagger-lemmatiser output. However,

[19]Tadpole is described in more detail in [49]. A more detailed account of how tagging and lemmatisation was actually applied in the case of the D-Coi pilot corpus is given in [48].

[20]At a later stage, another 500,000 words from the SoNaR corpus were manually corrected in the Lassy project. The total set of one million words is what we have elsewhere referred to as the SoNaR-1 corpus (cf. Sect. 3.4).

[21]FROG is available under GPL (online demo: http://ilk.uvt.nl/cgntagger/,software:http://ilk.uvt.nl/frog/). We refrained from applying FROG to data such as chats, tweets and SMS as we expected that FROG would perform very poorly on this type of data.

Table 13.3 Accuracy of Alpino on the manually corrected syntactically annotated part of D-Coi. The table lists the number of sentences, mean sentence length (in tokens), and F-score in terms of named dependencies

Corpus	Sentences	Length	F-score (%)
D-Coi	12,390	16	86.72

with the tool and procedure developed to support the manual verification of the data, users can yet undertake this task for specific subsets of the data as they see fit.

13.4.2 Syntactic Annotation

In the D-Coi project we also investigated the feasibility of (semi-)automatically annotating the corpus for syntactic information with Alpino, a computational analyzer of Dutch which was developed at the University of Groningen. Experiences with syntactic annotation in the Spoken Dutch Corpus (CGN) project had shown that the approach taken there was quite labour-intensive. Of course at the time of the CGN project, no syntactically annotated corpus of Dutch was available to train a statistical parser on, nor an adequate parser for Dutch.[22] However, at the start of the D-Coi project Alpino had sufficiently matured and became an option that deserved serious consideration while contemplating the syntactic annotation of large quantities of data.

Alpino provides full accurate parsing of unrestricted text and incorporates both knowledge-based techniques, such as a HPSG grammar and lexicon which are both organised as inheritance networks, as well as corpus-based techniques, for instance for training its disambiguation component. An overview of Alpino is given in [52]. Although the syntactic annotation scheme used by Alpino was based on the annotation guidelines that were developed earlier for the annotation of the Spoken Dutch Corpus, the annotation scheme deployed in D-Coi was not exactly the same as the one used in for the CGN [14, 42]. Differences include, for instance, the annotation of subjects of the embedded verb in auxiliary, modal and control structures, and the annotation of the direct object of the embedded verb in passive constructions. In the CGN scheme, these are not expressed. In D-Coi these subject relations are encoded explicitly.

Part of the pilot corpus (some 200,000 words) was annotated syntactically by means of Alpino and the annotations were manually corrected. In Table 13.3 we list the accuracy of Alpino on these data. With the syntactic annotations obtained by means of Alpino, we also inherited an XML format in which the syntactic

[22] An adequate parser should meet several requirements: it should have wide coverage, produce theory-neutral output, and provide access to both functional and categorial information.

annotations are stored. This format directly allows for the use of full XPath and/or Xquery search queries. As a result standard tools can be used for the exploitation of the syntactic annotations, and there is no need to dedicate resources to the development of specialised query languages.

After the D-Coi project was finished, syntactic annotation was further pursued in the STEVIN Lassy project. In this project, the one-million-word SoNaR-1 corpus was enriched with syntactic information. For more information we refer to Chap. 9, p. 147.

13.4.3 Annotation of Named Entities

Despite its huge application potential, the annotation of named entities and the development of named entity recognition (NER) systems is an under-researched area for Dutch. NER, the task of automatically identifying and classifying names in texts, has started as an information subtask in the framework of the MUC conferences, but has also been proven to be essential for information retrieval, question answering, co-reference resolution, etc.

The goal in the SoNaR project was to create a balanced data set labeled with named entity information, which would allow for the creation and evaluation of supervised machine learning named entity recognisers. The labeled data set substantially differs from the CoNLL-2002 shared task [45] data set, containing 309,686 tokens from four editions of the Belgian newspaper "De Morgen". First of all, the goal was to cover a wide variety of text types and genres in order to allow for a more robust classifier and better cross-corpus performance. Furthermore, instead of focusing on four named entity categories ("person", "location", "organisation" and "miscellaneous"), we aimed at a finer granularity of the named entities and we also wanted to differentiate between the literal and metonymic use of the entities. For the development of the guidelines, we took into account the annotation schemes developed in the ACE [11] and MUC (e.g. [4]) programmes, and the work on metonymy from [21]. In the resulting annotation guidelines, we focused on the delimitation of the named entities, after which each entity was potentially annotated with four annotation layers, covering its main type, subtype, usage and (in case of metonymic usage) its metonymic role (cf. Fig. 13.1).

The examples below clearly show that all tags maximally consist of four parts, in which the first part of the tag denotes the main type of the NE, the second part the sub type, the third one the use, and the last one the type of use.

1. Nederland[LOC.land.meto.human] gaat de bestrijding van het terrorisme anders en krachtiger aanpakken. Minister Donner[PER.lit] van justitie krijgt verregaande bevoegdheden in die strijd.
   (English: The Netherlands are planning to organise the fight against terrorism in a different and more powerful way. Minister of Justice Donner was given far-reaching powers in that battle.)

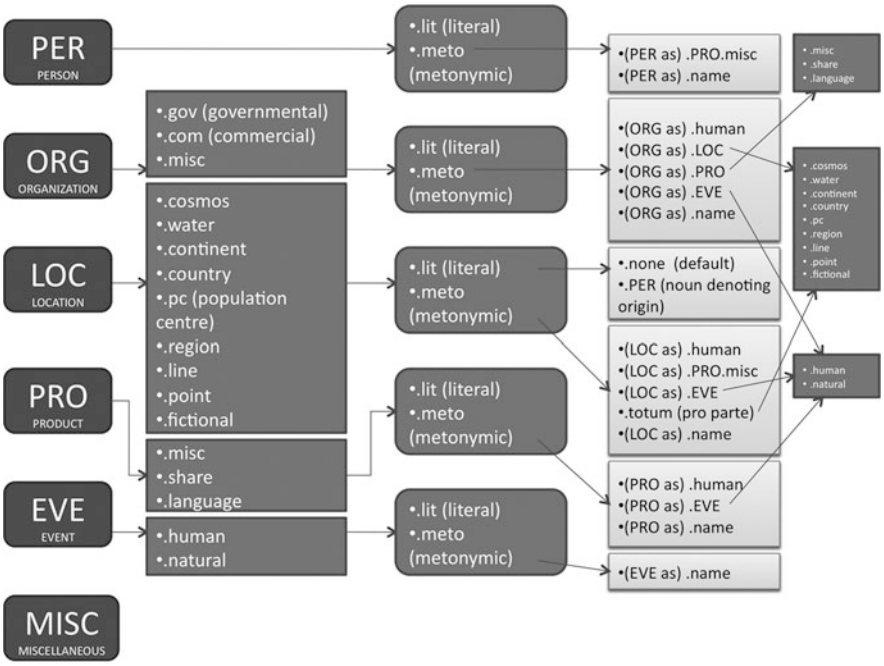


Fig. 13.1 Schematic overview of the named entity layers and the corresponding labels

2. Het is een eer om hier te zijn op MGIMO[ORG.misc.meto.loc]. Deze prachtige universiteit is een kweekvijver voor diplomatiek talent. Deze instelling heeft hechte contacten met Nederland[LOC.land.meto.human].
   (English: It is an honour to be here at MGIMO. This wonderful university is a breeding ground for diplomatic talent. This institution has tight connections with the Netherlands.)

The named entity annotations were performed on raw text and were done in the MMAX2[23] annotation environment. Annotation speed averaged around 3,500 words per hour. Taking into account the verification of the annotations by a second annotator, the actual annotation speed was about 2,000 words per hour. In order to evaluate the annotation guidelines, two annotators labeled eight randomly selected texts from the corpus (14,244 tokens in total). The interannotator agreement was measured with two evaluation metrics, namely Kappa [3] and F-measure ($\beta = 1$) [54]. The latter scores were calculated by taking one annotator as gold standard. The scores were calculated on five levels: span, main type, subtype, usage and metonymic role. For each level, scores were calculated on the entire set, and on a subset containing only those tokens on which both annotators agreed on the

[23] URL: http://mmax2.sourceforge.net

preceding level. For each of the levels, high agreement scores were obtained, with a Kappa score ranging from 0.97 to 0.91 and an F-score ranging from 99.6 to 98.9 %. For a detailed description of the guidelines and the interannotator agreement on each of the annotation levels, we refer to [10].

The annotated corpus was used for the development of a NE classifier [10], which was used for the automatic annotation of the remaining 499 million words. Although the one-million-word corpus already covered different text types, thus allowing to have a more balanced view on the quality of the named entity recogniser, this does not guarantee that the automatic labeling of the 499 million remaining words reaches the same accuracy levels. We expect that an adaptation of the classifier to informal text types (blogs, chats, sms) will be required. In order to allow for this adaptation, the full named entity recogniser was also delivered together with the manually verified annotations.

13.4.4 Annotation of Co-reference Relations

In the last decade, considerable efforts have been put in annotating corpora with co-referential relations in order to support the development of co-reference resolution systems. Co-reference resolution is the task of automatically recognising which words or expressions (most often noun phrases) refer to the same discourse entity in a particular text or dialogue. The applicability of the accurate identification of co-reference relations between noun phrases is huge: in information extraction, question answering or in machine translation. Therefore, not only a widespread language such as English (e.g. ACE-2 [11], ARRAU [30], OntoNotes 3.0 [56]), but also smaller languages such as Czech (PDT 2.0; [19]) and Catalan (AnCora-Ca; [31]) can now rely on annotated resources for co-reference resolution. Through the annotation of the SoNaR-1 corpus, we created one of the largest data sets currently available to co-reference resolution research. Furthermore, the balanced nature of the data also allows for studying cross-genre performance [9].

The first Dutch corpus annotated with co-referential relations between nominal constituents was created in 2005 [15]. In the STEVIN COREA project, the annotation guidelines from [15] were refined and also extended to the labeling of bridge relations [12].[24] These COREA guidelines served as the basis for the annotation of co-reference in the SoNaR-1 corpus. The guidelines allow for the annotation of four relations and special cases are flagged. The four annotated relations are identity (NPs referring to the same discourse entity), bound, bridge (as in part-whole, superset-subset relations) and predicative. The following special cases were flagged: negations and expressions of modality, time-dependency and identity of sense (as in the so-called paycheck pronouns [18]). Co-reference links

[24] See also Chap. 7, p. 115.

were annotated between nominal constituents, which could take the form of a pronominal, named entity or common noun phrase, as exemplified in (3), (4) and (5).

3. Nederland gaat de bestrijding van het terrorisme [id="21"] anders en krachtiger aanpakken. Minister Donner van justitie krijgt verregaande bevoegdheden in die strijd [id = "2" ref="1" type="ident"].
4. Het is een eer om hier te zijn op MGIMO [id="1"]. Deze prachtige universiteit [id="2" ref="1" type="ident"] is een kweekvijver voor diplomatiek talent [id="3" ref="1" type="pred"]. Deze instelling [id="4" ref="1" type="ident"] heeft hechte contacten met Nederland.
5. Binnen in de gymzaal [id="1"] plakken gijzelaars [id="2"] de ramen [id="3" ref="1" type="bridge"] af en plaatsen ze [id="4" ref="2" type="ident"] explosieven aan de muur [id="5" ref="1" type="bridge"].
   (English: Inside the gym, the hijackers covered the windows and attached explosives to the walls)

In order to avoid conflicts between the annotation layers, the co-reference annotations were performed on the nominal constituents, which were extracted from the manually validated syntactic dependency trees [53]. Furthermore, we checked for inconsistencies with the named entity layer. We again used MMAX2 as annotation environment.

Since inter-annotator agreement for this labeling task was already measured in the framework of the design of the annotation guidelines [12], no separate inter-annotator agreement assessment was done. Hendrickx et al. [12] computed the inter-annotator agreement on the identity relations as the F-measure of the MUC-scores [55] obtained by taking one annotation as 'gold standard' and the other as 'system output'. They report an inter-annotator agreement of 76 % F-score on the identity relations. For the bridging relations, an agreement of 33 % was reported.

Due to the low performance of the current classification-based co-reference resolution systems for Dutch [12, 15] no automatic pre-annotation was performed to support or accelerate the annotation process.

13.4.5 Annotation of Semantic Roles

The labeling of semantic roles was initiated in the D-Coi project and resulted in a set of guidelines [46] which were further extended in the SoNaR project and a small labeled data set of about 3,000 predicates. For the development of the guidelines, we considered the annotation scheme proposed within existing projects such as FrameNet [17] and PropBank [29]. Mainly because of the promising results obtained for automatic semantic role labeling using the PropBank annotation scheme, we decided to adapt the latter scheme to Dutch. In the case of traces, PropBank creates co-reference chains for empty categories while in our case, empty categories are almost non-existent and in those few cases in which they are attested, a co-indexation has been established already at the syntactic level. Furthermore,

in SoNaR we assume dependency structures for the syntactic representation while PropBank employs phrase structure trees. In addition, Dutch behaves differently from English with respect to certain constructions (i.e. middle verb constructions) and these differences were also spelled out.

Besides the adaptation (and extension) of the guidelines to Dutch, a Dutch version of the PropBank frame index was created. In PropBank, frame files provide a verb specific description of all possible semantic roles and illustrate these roles by examples. The lack of example sentences makes consistent annotation difficult. Since defining a set of frame files from scratch is very time consuming, we annotated Dutch verbs with the same argument structure as their English counterparts, thus using English frame files instead of creating Dutch ones.

For the annotation of the semantic roles, we relied on the manually corrected dependency trees and TrEd[25] was used as annotation environment.

The PropBank role annotation is exemplified below, using two previously introduced examples (cf. (3) and (5)):

6. Nederland(Arg0)— gaat — de bestrijding van het terrorisme (Arg1) — anders en krachtiger (ArgM-MNR) — aanpakken (PRED). Minister Donner van justitie (Arg0)— krijgt (PRED) — verregaande bevoegdheden in die strijd (Arg1).
7. Binnen in de gymzaal (ArgM-LOC) — plakken (PRED) — gijzelaars (Arg0) — de ramen (Arg1) — af en —plaatsen (PRED)— ze (Arg0) —explosieven(Arg1)— aan de muur (Arg2).

Lacking a training corpus for Duch semantic role labeling, we initially created a rule-based tagger based on D-Coi dependency trees [24], called XARA (XML-based Automatic Role-labeler for Alpino-trees). It establishes a basic mapping between nodes in a dependency graph and PropBank roles. A rule in XARA consist of an XPath expression that addresses a node in the dependency tree, and a target label for that node, i.e. a rule is a (path, label) pair. Once sufficient training data were available, we also developed a supervised classifier, and more specifically the memory-based learning classifiers implemented in TiMBL [6], for the task. Instead of starting annotation from scratch we decided to train our classifier on the sentences annotated for D-Coi in order to pre-tag all sentences, thus rephrasing the annotation task as a verification task. After manually verifying 50,000 words we performed a first error analysis and retrained the classifier on more data in order to bootstrap the annotation process. In total, 500,000 words were manually verified. This dataset again served as the basis for the further adaptation of the classifier, which also takes into account the results of the new annotation layers of NE and co-reference. This adapted classifier labeled the remaining 500K of the SoNaR-1 corpus.

[25] URL: http://ufal.mff.cuni.cz/$\sim$pajas/tred/o

13.4.6 Annotation of Temporal and Spatial Entities

Whereas usually these two layers of annotation are handled separately, we have used STEx (which stands for Spatio Temporal Expressions), a combined spatiotemporal annotation scheme. STEx takes into account aspects of both TimeML [36] upon which the recent ISO standard ISO TimeML is mainly based[26] and SpatialML[43], serving as an ISO standard under construction. A first version of STEx, MiniSTEx, was developed within the D-Coi project, the tool used there being a semi-automatic one. Work on MiniSTEx was continued in the AMASS++-project (IWT-SBO). The resulting STEx approach is a hybrid one, which uses rules, a large spatio-temporal knowledge base, the Varro toolkit (cf. [22, 23]) and TiMBL [7] to annotate texts fully automatically. The correctors are not confronted with tags with an under-treshold probability in case several tags are in se possible unless all of these are under-treshold.

Within the SoNaR project, the STEx spatial scheme was largely restricted to geospatial annotation.[27] Moreover, due to financial and temporal restrictions, we had to limit ourselves to recognition and normalisation of temporal and geospatial entities, while reasoning was ignored.

The current STEx scheme handles spatial and temporal expressions much in the same way as MiniSTEx [37–39], i.e., contrary to ISO TimeML and (ISO) SpatialML, in combination (cf. Table 13.4). We consider this quite a unique characteristic of our approach [41]. Another point in which STEx deviates from other approaches concerns the use of a feature noise. People often formulate carelessly, even journalists in quality newspapers or weeklies, for example mixing Engels (English) and Brits (British) in "de Engelse/Britse minister-president". As England is in Great Britain, would this mean that there are two prime-ministers, one of England and one of Great Britain? Or is this to be considered noisy information as in Dutch the notions England, United Kingdom and Great Britain are often mixed up? And when someone remarked the 30th of April 2011 to have been in Paris a year ago, does that mean that person was there the 30th of April 2010 (on the exact date) or rather that he or she was there around that date? In STEx such expressions come with a feature noise = "yes".

Besides the fact that STEx uses geospatial information to determine temporal information and the other way around, STEx also differs from both TimeML and SpatialML in that it is provides more details (cf. [38, 39]). In the AMASS++-project this turned out to be very useful in multidocument applications, like summarisation and information retrieval as it makes available information not expressed in a text.

8. Zij hebben hun zoon gisteren [temp type="cal" ti="tp-1" unit="day" val= "2008-05-22"] in Amsterdam [geo type="place" val="EU::NL::-::NH::

[26] Cf. TimeML Working Group 2010.

[27] In the ISO working group on SpatialML most attention up till now was devoted to spatial phenomena in general, not to geospatial ones.

Table 13.4 The resemblance between temporal and spatial analyses

Temporal	Geospatial
Time of perspective	Place of perspective
Time of location	Place of location
Time of eventuality	Place of eventuality
Duration	Distance
Shift of perspective	Shift of perspective
Relations	Relations

Amsterdam::Amsterdam" coord="52.37,4.9"] gezien [temp type="event" value="vtt" rel="before(ti,tp)"]
(English: They saw their son yesterday in Amsterdam)

In example (8) the time-zone associated with it (timezone = "UTF+1") is filtered out, although it is contained in the metadata coming with the text. Only when its value is overruled by a statement in the text it will be mentioned in the annotation itself. Example (8) also contains a shorthand version of the formulas we associated with several temporal expressions. ti = "tp-1" unit = "day" says that the time of eventuality ti is the time of perspective tp minus 1. As the unit involved is that of day, only that variable is to be taken into account. So, yesterday is to be associated with a formula, not with an accidental value (like "2008-05-22" in (8)). In a second step, the calculations are to be performed. This is crucial for a machine learning approach: not the value for yesterday is to be learned, but the formula associated with it.

In the context of the SoNaR corpus, STEx made use of the information available through previous syntactic and semantic layers.[28,29] In some cases it completed and disambiguated such information. For example, the location related annotations at the level of NER would be disambiguated. When a sentence like (8) occurred in a document, usually an expression like Amsterdam could be disambiguated, stating that the instantiation of Amsterdam meant was the town of Amsterdam in the Netherlands, not one of the towns or villages in the US, Canada, ... Especially in a corpus, the metadata coming with a file allow for such an annotation (cf. [38]). Co-reference was also very useful, the same holds especially for metonymy as annotated in NER (cf. also [20]). As remarked above, spatio-temporal annotation in SoNaR was performed (semi-)automatically, using a large knowledge base containing geospatial and temporal data, combinations of these and especially also cultural data with respect to such geospatial and temporal data. Cultural aspects like tradition (Jewish, Christian), geographical background, social background have their effects on the (intended) interpretation of temporal and geospatial data (cf. Fig. 13.2) by

[28] With regard to the exception of the Semantic Role Labeling (SRL) which was ignored, as for practical reasons SRL and STEx were performed in parallel.

[29] In the AMASS++ project [40] a version of STEx was used in which it had to rely on automatic PoS tagging and chunking. In a future paper we intend to compare such approaches: is manual correction/addition of further layers of annotation worth the effort (time and money)?

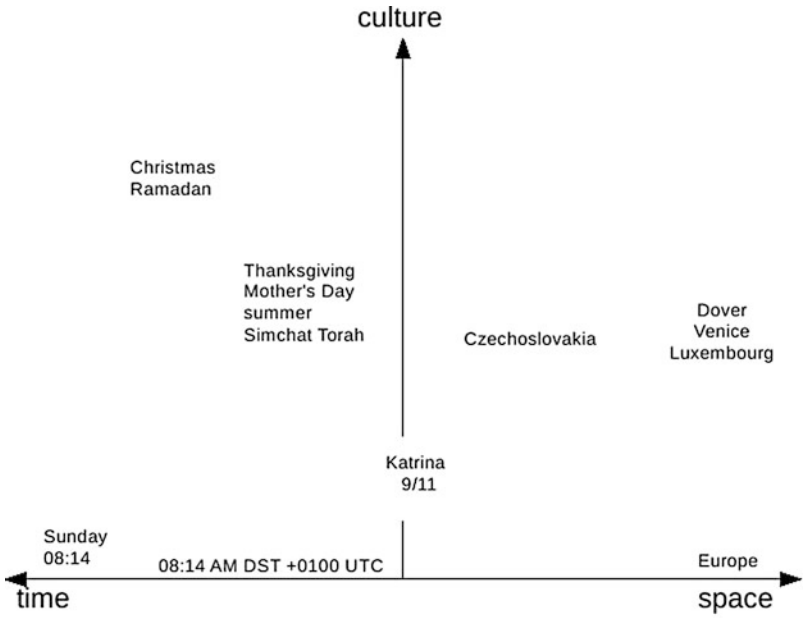


Fig. 13.2 Eventualities with temporal, geospatial or and/or cultural aspects

the people meant to read a specific text. For example: what is considered as the begin and end dates of World War II is not the same all over Europe and the rest of the world.[30] The same holds for the date(s) associated with Christmas, or Thanksgiving. Or to decide which Cambridge (UK, US) is referred to, or which Antwerpen (Antwerp): the province, the municipality or the populated place.[31] Each annotation was in principle corrected by one corrector (student), some substantial parts were corrected by more students in order to ensure annotator agreement. The time needed for correcting a file depended on the type of file, even on its topic. Legal texts for example, we found, were rather easy. However, the description of the history of a few Dutch hamlets over the last 500 years or the ins and outs of the American Civil War might take very long as in those cases the knowledge base will not contain all the relevant data.

[30] With regard to begin date: September 1939 (invasion of Poland), May 1940 (invasion of The Netherlands and Belgium), December 1941 (US, Pearl Harbor). Or . . . ?

[31] At the moment, the precision for such geospatial anchors in STEx is 0.92, recall 0.91 (small scale test for some 200 instances).

13.5 Concluding Remarks

While the Spoken Dutch Corpus already provided researchers with spoken language data, at the start of the STEVIN programme the dire need for a large resource for written data persisted. Through investment in the D-Coi and SoNaR projects directed at the construction of a 500 MW corpus an important gap in the Dutch language resources infrastructure was filled. But the impact of these projects extends well beyond the delivery of the 500 MW reference corpus as significant contributions were made to the development and consolidation of de facto standards, and tools and procedures were developed that were also used in various other projects.[32]

Although the D-Coi project was defined as a preparatory project which aimed to develop the procedures, protocols and tools needed for the construction of a large corpus, one of the more tangible results for the end-user was the 54 MW pilot corpus that was compiled [26]. In order to facilitate corpus exploitation, COREX – the corpus exploitation software developed for use with the Spoken Dutch Corpus – was adapted so that with one and the same tool both the Spoken Dutch corpus and the D-Coi corpus can now be accessed. The D-Coi corpus and the exploitation software are available through the Dutch HLT Agency.[33]

Through the SoNaR project two further corpora have become available: the SoNaR-500 corpus and the SoNaR-1 corpus. The SoNaR-500 corpus is available in two formats, the D-Coi+ format and the latest development FoLiA (Format for Linguistic Annotation; [51]). With the D-Coi+ format we are compatible with previous (intermediate) releases of the corpus. However, as the D-Coi+ format is not capable of accommodating the annotations for NE and has no provisions for specific characteristics associated with data from the new media, we have decided to adopt FoLiA for which this is not a problem. The annotations for the SoNaR-1 corpus are available in the formats as they were produced, i.e. MMAX for co-reference and named entities, TrEd for semantic roles, STEx XML for temporal and spatial entities.

For the exploitation of the 500 MW reference corpus presently no exploitation software is available, nor is the development of such software presently foreseen. For the exploitation of the SoNaR-1 corpus dedicated tools are already available for the syntactic annotation (cf. Chap. 9, p. 147), while currently in the context of

[32] Standards developed in D-Coi and SoNaR have been used in for example the STEVIN Jasmin-CGN and Dutch Parallel Corpus projects but also in the NWO-funded BasiLex and Dutch SemCor projects. As for tools and procedures, the corpus clean-up procedure developed by Reynaert has been adopted in the NWO-funded Political Mashup project and a project funded by CLARIN-NL, viz. VU-DNC, while it is also available as a web application/servide in the CLARIN infrastructure. Experiences in the D-Coi project have guided the development of by now widely used tools such as the Tilburg tagger/lemmatisers and the Alpino parser.

[33] With additional funds from NWO the HLT Agency together with Polderland Language and Speech Technology bv continued to develop the tool. The aim was to make corpora accessible over the internet and to make possible the exploitation of other corpora (such as JASMIN-CGN).

the TTNWW project all the tools and the semantic annotations discussed in this chapter will be made more easily accessible, especially for researchers in human and social sciences.[34] Apart from the D-Coi pilot corpus and the SoNaR-500 and the SoNaR-1 corpora, there are large quantities of surplus materials. As observed in Sect. 13.2.2, to the extent possible within the limitations of the SoNaR project, these data have been processed. Of the materials that presently remain in their original form a substantial part is in PDF. In our experience it is advisable to leave these data be until such a time when at some point in the future there is a breakthrough in the text extraction technology which makes it possible to extract text from PDF without losing valuable information.[35]

Acknowledgements Thanks are due to our collaborators in these projects (in random order): Paola Monachesi, Gertjan van Noord, Franciska de Jong, Roeland Ordelman, Vincent Vandeghinste, Jantine Trapman, Thijs Verschoor, Lydia Rura, Orphée De Clercq, Wilko Apperloo, Peter Beinema, Frank Van Eynde, Bart Desmet, Gert Kloosterman, Hendri Hondorp, Tanja Gaustad van Zaanen, Eric Sanders, Maaske Treurniet, Henk van den Heuvel, Arjan van Hessen, and Anne Kuijs.

Appendix

In the first column of Table A.1 the various corpus components and text types are listed. The second column indicates the data volumes foreseen in the original design. The third column shows the data volumes in the D-Coi pilot corpus. The remaining three columns give the data volumes actually realised in the SoNaR-500 corpus. NLD stands for data originating from the Netherlands, BEL for data from Flanders, and OTH for data whose origin could not be established. Data volumes are in millions of words.

[34]The acronym TTNWW stands for TST Tools voor het Nederlands als Webservices in een Workflow (HLT Tools for Dutch as Web Services in a Work Flow). This Flemish-Dutch pilot project is financed by the Flemish (Department of Economy, Science and Innovation) and Dutch (via CLARIN-NL) governments.

[35]For a recent appraisal of the state of the art in PDF text extraction technology we refer to a recent technical paper released by Mitre [13]. The main conclusion there is that all too often valuable textual information is irretrievably lost when extracting text from PDF even when one uses the currently best-of-breed PDF text extractor available.

Table A.1

			SoNaR-500		
	Original design	D-Coi	NLD	BEL	OTH
Written to be read, published, electronic	177.5	27.3	36.8	59.2	32.8
Written to be read, published, printed	185.0	25.1	101.4	233.9	19.5
Written to be read, unpublished, electronic	100.0	0	1.6	11.4	0
Written to be read, unpublished, printed	10.0	0	0	0	0
Written to be read, unpublished, typed	20.0	0	0	0	0
Written to be spoken, unpublished, electronic	2.5	0.9	2.8	25.3	0
Written to be spoken, unpublished, typed	5.0	0.7	0.7	0	0

References

1. Aston, G., Burnard, L.: The BNC Handbook. Exploring the British National Corpus with SARA. Edinburgh University Press, Edinburgh (1998)
2. Braasch, A., Farse, H., Jongejan, B., Navaretta, C., Olsen, S., Pedersen, B.: Evaluation and Validation of the D-Coi Pilot Corpus. Center for Sprokteknologi, Copenhagen (2008)
3. Carletta, J.C.: Assessing agreement on classification tasks: the kappa statistic. Comput. Linguist. **22**(2), 249–254 (1996)
4. Chinchor, N., Robinson, P.: MUC-7 Named Entity Task Definition (version 3.5) (1998)
5. Daelemans, W., Strik, H.: Het Nederlands in de taal-en Spraaktechnologie: prioriteiten Voor Basisvoorzieningen. Nederlandse Taalunie, The Hague (2002)
6. Daelemans, W., van den Bosch, A.: Memory-Based Language Processing. Cambridge University Press, Cambridge (2005)
7. Daelemans, W., Zavrel, J., van der Sloot, K., van den Bosch, A.: TiMBL: tilburg memory based learner, version 5.1.0, reference guide. Technical Report ILK 04-02, ILK Research Group, Tilburg University (2004)
8. De Clercq, O., Reynaert, M.: SoNaR acquisition manual version 1.0. Technical Report LT3 10-02, LT3 Research Group – Hogeschool Gent (2010). http://lt3.hogent.be/en/publications/
9. De Clercq, O., Hoste, V., Hendrickx, I.: Cross-domain Dutch coreference resolution. In: Proceedings of the 8th International Conference on Recent Advances in Natural Language Processing. RANLP 2011, Hissar, Bulgaria (2011)

10. Desmet, B., Hoste, V.: Named entity recognition through classifier combination. In: Computational Linguistics in the Netherlands 2010: Selected Papers from the Twentieth CLIN Meeting, Utrecht (2010)
11. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, R., Strassel, S., Weischedel, R.: The automatic content extraction (ACE) program tasks, data, and evaluation. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation, Lisbon, pp. 837–840. LREC-2004 (2004)
12. Hendrickx, I., Bouma, G., Coppens, F., Daelemans, W., Hoste, V., Kloosterman, G., Mineur, A.M., Vloet, J.V.D., Verschelde, J.L.: A coreference corpus and resolution system for Dutch. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation, Marrakech, pp. 144–149. LREC-2008 (2008)
13. Herceg, P.M., Ball, C.N.: A comparative study of PDF generation methods: measuring loss of fidelity when converting Arabic and Persian MS Word files to PDF. Technical Report MTR110043, Mitre (2011). http://www.mitre.org/work/tech_papers/2011/11_0753/11_0753.pdf
14. Hoekstra, H., Moortgat, M., Renmans, B., Schouppe, M., Schuurman, I., Van der Wouden, T.: CGN syntactische annotatie. http://www.ccl.kuleuven.be/Papers/sa-man_DEF.pdf (2004)
15. Hoste, V.: Optimization issues in machine learning of coreference resolution. Ph.D. thesis, Antwerp University (2005)
16. Ide, N., Macleod, C., Fillmore, C., Jurafsky, D.: The American national corpus: an outline of the project. In: Proceedings of International Conference on Artificial and Computational Intelligence. ACIDCA-2000, Monastir (2000)
17. Johnson, C.R., Fillmore, C.J., Petruck, M.R.L., Baker, C.F., Ellsworth, M.J., Ruppenhofer, J., Wood, E.J.: FrameNet: theory and practice. ICSI Technical Report tr-02-009 (2002)
18. Karttunen, L.: Discourse Referents. Syntax and Semantics, vol. 7. Academic, New York (1976)
19. Kučova, L., Hajičova, E.: Coreferential relations in the Prague dependency treebank. In: Proceedings of DAARC 2004, Azores, pp. 97–102 (2004)
20. Leveling, J., Hartrumpf, S.: On metonymy recognition for geographic information retrieval. Int. J. Geogr. Inf. Sci. **22**(3), 289–299 (2008)
21. Markert, K., Nissim, M.: Towards a corpus annotated for metonymies: the case of location names. In: Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas, pp. 1385–1392. LREC-2002 (2002)
22. Martens, S.: Varro: an algorithm and toolkit for regular structure discovery in treebanks. In: Proceedings of Coling 2010, Beijing, pp. 810–818 (2010)
23. Martens, S.: Quantifying linguistic regularity. Ph.D. thesis, KU Leuven (2011)
24. Monachesi, P., Stevens, G., Trapman, J.: Adding semantic role annotation to a corpus of written Dutch. In: Proceedings of the Linguistic Annotation Workshop (Held in Conjunction with ACL 2007), Prague (2007)
25. Oostdijk, N.: The spoken dutch corpus. Outline and first evaluation. In: Proceedings of the Second International Conference on Language Resources and Evaluation, Athens, pp. 887–894. LREC-2000 (2000)
26. Oostdijk, N.: Dutch language corpus initiative, pilot corpus. Corpus description. TR-D-COI-06-09 (2006)
27. Oostdijk, N.: A reference corpus of written Dutch. Corpus design. TR-D-COI-06f (2006)
28. Oostdijk, N., Boves, L.: User requirements analysis for the design of a reference corpus of written Dutch. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, Genoa, pp. 1206–1211. LREC-2006 (2006)
29. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: a corpus annotated with semantic roles. Comput. Linguist. J. **31**(1) (2005)
30. Poesio, M., Artstein, R.: Anaphoric annotation in the ARRAU corpus. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation, Marrakech, pp. 1170–1174. LREC-2008 (2008)
31. Recasens, M., Marti, M.A.: AnCora-CO: coreferentially annotated corpora for Spanish and Catalan. Lang. Resour. Eval. **44**(4), 315–345 (2010)

32. Reynaert, M.: Corpus-induced corpus cleanup. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC-2006, Trento, pp. 87–92 (2006)
33. Reynaert, M.: Non-interactive OCR post-correction for giga-scale digitization projects. In: Gelbukh, A. (ed.) Proceedings of the Computational Linguistics and Intelligent Text Processing 9th International Conference, CICLing 2008, vol. 4919, pp. 617–630. Springer, Berlin (2008)
34. Reynaert, M.: Character confusion versus focus word-based correction of spelling and OCR variants in corpora. Int. J. Doc. Anal. Recognit. 1–15 (2010). http://dx.doi.org/10.1007/s10032-010-0133-5, doi:10.1007/s10032-010-0133-5
35. Sanders, E.: Collecting and analysing chats and tweets in SoNaR. In: Proceedings of the Eighth International Conference of Language Resources and Evaluation, Istanbul, pp. 2253–2256. LREC-2012 (2012)
36. Sauri, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., Pustejovsky, J.: TimeML annotation guidelines, version 1.2.1. http://timeml.org/site/publications/specs.html (2006)
37. Schuurman, I.: Spatiotemporal annotation on top of an existing treebank. In: De Smedt, K., Hajic, J., Kuebler, S. (eds.) Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories, Bergen, pp. 151–162 (2007)
38. Schuurman, I.: Which New York, which Monday? The role of background knowledge and intended audience in automatic disambiguation of spatiotemporal expressions. In: Proceedings of CLIN 17, Leuven (2007)
39. Schuurman, I.: Spatiotemporal annotation using MiniSTEx: how to deal with alternative, foreign, vague and obsolete names? In: Proceedings of the Sixth Conference on International Language Resources and Evaluation (LREC'08), Marrakech (2008)
40. Schuurman, I., Vandeghinste, V.: Cultural aspects of spatiotemporal analysis in multilingual applications. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta (2010)
41. Schuurman, I., Vandeghinste, V.: Spatiotemporal annotation: interaction between standards and other formats. In: IEEE-ICSC Workshop on Semantic Annotation for Computational Linguistic Resources, Palo Alto (2011)
42. Schuurman, I., Schouppe, M., Van der Wouden, T., Hoekstra, H.: CGN, an annotated corpus of spoken Dutch. In: Proceedings of the Fourth International Conference on Linguistically Interpreed Corpora, Budapest, pp. 101–112. LINC-2003 (2003)
43. SpatialML: Annotation Scheme for Marking Spatial Expressions in Natural Language. MITRE (2007). Version 2.0, LDC, Upenn
44. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., Varga, D.: The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, Genoa, pp. 2142–2147. LREC-2006 (2006) http://arxiv.org/ftp/cs/papers/0609/0609058.pdf
45. Tjong Kim Sang, E.: Introduction to the CoNLL-2002 shared task: language-independent named entity recognition. In: Proceedings of the 6th Conference on Natural Language Learning, Taipei, pp. 155–158 (2002)
46. Trapman, J., Monachesi, P.: Manual for semantic annotation in D-Coi. Technical Report, Utrecht University (2006)
47. Treurniet, M., De Clercq, O., Oostdijk, N., Van den Heuvel, H.: Collecting a corpus of Dutch SMS. In: Proceedings of the Eighth International Conference of Language Resources and Evaluation, Istanbul, pp. 2268–2273. LREC-2012 (2012)
48. Van den Bosch, A., Schuurman, I., Vandeghinste, V.: Transferring PoS-tagging and lemmatisation tools from spoken to written Dutch corpus development. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, Genoa. LREC-2006 (2006)
49. Van den Bosch, A., Busser, B., Canisius, S., Daelemans, W.: An efficient memory-based morphosyntactic tagger and parser for Dutch. In: Dirix, P., Schuurman, I., Vandeghinste, V., Van Eynde, F. (eds.) Computational Linguistics in the Netherlands: Selected Papers from the Seventeenth CLIN Meeting, Leuven, pp. 99–114 (2007)

50. Van Eynde, F.: Part of speech tagging en lemmatisering. Protocol voor annotatoren in D-Coi. Centrum voor Computerlinguïstiek, Leuven. http://www.let.rug.nl/vannoord/Lassy/POS-manual.pdf internal document
51. van Gompel, M.: Folia: format for linguistic annotation. http://ilk.uvt.nl/folia/folia.pdf (2011)
52. Van Noord, G.: At last parsing is now operational. In: Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues Naturelles, Leuven, pp. 20–42. TALN-2006 (2006)
53. Van Noord, G., Schuurman, I., Vandeghinste, V.: Syntactic annotation of large corpora in STEVIN. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, Genoa, pp. 1811–1814. LREC-2006 (2006)
54. Van Rijsbergen, C.: Information Retrieval. Buttersworth, London (1979)
55. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Proceedings of the Sixth Message Understanding Conference (MUC-6), Columbia, pp. 45–52 (1995)
56. Weischedel, R., Pradhan, S., Ramshaw, L., Palmer, M., Xue, N., Marcus, M., Taylor, A., Greenberg, C., Hovy, E., Belvin, R., Houston, A.: OntoNotes Release 3.0. LDC2009T24. Linguistic Data Consortium (2009)
57. Woordenlijst Nederlandse Taal: SDU Uitgevers, The Hague (1995)

Part III
HLT-Technology Related Papers

Chapter 14
Lexical Modeling for Proper name Recognition in Autonomata Too

Bert Réveil, Jean-Pierre Martens, Henk van den Heuvel, Gerrit Bloothooft, and Marijn Schraagen

14.1 Introduction

Points of Interest business applications are strongly emerging on the ICT market, in particular in high-end navigation systems. For cars for instance, there is a high safety issue, and voice-driven navigation systems are appealing because they offer hands- and (partly) eye-free operation. However, Points of Interest like company names, hotel and restaurant names, names of attraction parks and museums, etc., often contain non-native parts. Moreover, the targeted application must be usable by non-native as well as native speakers. This means that there are considerable cross-lingual effects to cope with, which implies that the challenges for the automatic recogniser are high. At the start of the project (February, 2008) there was indeed substantial evidence [1–8] that state-of-the-art ASR technology was not yet good enough to enable a sufficiently reliable voice-driven POI business service.

The general project aim was therefore to improve name recognition accuracy by better coping with the large degree of variations observed in the POI pronunciations. The specific aim was to improve the recognition of (1) native Dutch/Flemish pronunciations of Dutch/Flemish POI, (2) native Dutch/Flemish pronunciations of foreign POI, and (3) non-native pronunciations of Dutch and Flemish POI. An important constraint was that the envisaged approach would have to be easily transferable from one application domain (e.g. car navigation) to another (e.g.

B. Réveil (✉) · J.-P. Martens
Ghent University, ELIS-DSSP, Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium
e-mail: bert.reveil@elis.ugent.be; martens@elis.ugent.be

H. van den Heuvel
Radboud University, CLST, Erasmusplein 1, 6500 HD Nijmegen, The Netherlands
e-mail: H.vandenHeuvel@let.ru.nl

G. Bloothooft · M. Schraagen
Utrecht Institute of Linguistics, OTS, Trans 10, 3512 JK Utrecht, The Netherlands
e-mail: g.bloothooft@uu.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_14, © The Author(s) 2013

telephone-based services for ordering medication, whiskey brands, etc.). Therefore, we contemplated an approach that would require no transcribed spoken name utterances from the targeted application domain. The domain knowledge would have to be provided in the form of example phonemic name transcriptions that are easy to acquire from people who know the application domain. The method would initially be developed and assessed for the domain of person name and geographical name recognition because for this domain transcribed utterances were available for development and evaluation, thanks to the Autonomata project (cf. Chap. 4, p. 61 on Autonomata resources). Subsequently, it would be transferred to the domain of POI recognition for which no transcribed development data were available yet.

The general concept of our methodology is that new pronunciation variants are generated with so-called P2P converters that apply automatically learned context-dependent transformation rules on the output of general-purpose G2P converters.

The rest of this chapter is organised as follows. In Sect. 14.2 we give a survey of multilingual pronunciation and acoustic modeling methods that were previously proposed for improving proper name recognition. In that section we also discuss the experiments that we conducted in order to define a state-of-the-art baseline system. In Sect. 14.3 we try to quantify how much further improvement is possible by means of more advanced pronunciation modeling techniques. In Sect. 14.4 we discuss the approach that we developed and the elements that make it unique. In Sect. 14.5 we offer an experimental validation of our method in the person and geographical name domains as well as in the targeted POI domain. The main conclusions of our work are formulated in Sect. 14.6.

14.2 Formerly Proposed Approaches

It has been shown by many authors that when cross-lingual factors come into play both acoustic and lexical modeling techniques can help to improve the ASR accuracy. For proper name recognition this is evidently the case, which is why we briefly review some of these techniques and why we assessed them when applied to proper name recognition.

14.2.1 Acoustic Modeling Approaches

Acoustic modeling tries to cope with the different ways in which an intended sound (a phoneme) can be articulated by the speaker. For the particular case of accented speech, a well known recipe to improve the recognition is to collect a small accented speech corpus and to adapt native acoustic models to the considered accent on the basis of this corpus. Popular adaptation methods in this respect are maximum likelihood linear regression (MLLR) [9] and maximum a posteriori (MAP) adaptation [10]. In [11], this technique yielded a 25 % improvement for the recognition of

English text spoken by Japanese natives with a low-proficiency in English. In [12], MLLR and MAP adaptation were used sequentially to adapt context-independent native acoustic models to an a priori known accent. Improvements of over 50 % could be attained in the context of an automated vocal command system.

An alternative approach is to start with a multilingual phoneme set and multilingual training data and to train context-dependent phoneme models on data from all languages in which the corresponding phonemes appear. By doing so for a bilingual set-up (German as native and English as foreign language), [13] could improve the recognition of (partly) English movie titles read by German natives by 25 % relative. In [14], the problem of recognising accented English speech embedded in Mandarin speech is tackled. Improvements of around 20 % relative over the standard multilingual approach were obtained by merging the output distribution of each bilingual model state with that of a related Mandarin accented English model state. The related state is identified automatically using a measure of the acoustic distance between states.

In [15,16], the more challenging case of multiple foreign accents was considered. French commands and expressions uttered by speakers from 24 different countries were recognised using a baseline French system, and a multilingual system that was obtained by supplementing the French acoustic models with three foreign (English, German and Spanish) acoustic model sets that were trained on speech from the corresponding languages. The multilingual acoustic models did improve the recognition for English and Spanish speakers (by about 15–20 %), but unexpectedly, degraded it for German speakers (by about 25 %). Furthermore, there was also a significant degradation for native French speakers and non-native French speakers of non-modeled languages.

14.2.2 Lexical Modeling Approaches

Lexical modeling deals with the phonetisation process, defined as the internal conversion of the orthography to a phonemic transcription that then serves as the basis for the articulation. It is generally known that non-native speakers often perform a non-standard phonetisation. In order to deal with this phenomenon, lexical modeling tries to enrich a baseline lexicon with the most frequently occurring non-standard phonetisations. One popular recipe is to add transcriptions emerging from G2P converters that implement the phonetisation rules of the most relevant foreign languages. In [1], Dutch, English and French G2P transcriptions were included for all entries (about 500) in a pronunciation dictionary containing Dutch, English, French and other names. Using optimised language dependent weights for the transcriptions, the name error rate could be reduced by about 40 % for native Dutch speakers, 70 % for French speakers, 45 % for English speakers and 10 % for other foreign speakers.

A similar approach was adopted in [6], but in a larger scale set-up with a vocabulary of 44K person names that occur in the US. Two baseline pronunciation

dictionaries were constructed: one with handcrafted typical native US English transcriptions (TY) and one with transcriptions emerging from a native US English G2P converter. Then, new variants were generated by eight foreign G2P converters covering all foreign language origins of the names occurring in the data set. Using n-gram grapheme models as language identifiers, likelihoods for the name source languages were computed and the transcriptions generated by the top two foreign G2P converters were added to the baseline lexicons. The variants caused a 25 % reduction of the name error rate for all names uttered by non-native speakers, irrespective of the baseline lexicon. However, the error rate reduction was only 10 % for the native utterances of foreign names and insignificant for the native utterances of native names.

14.2.3 Assessment of Established Approaches

In order to assess the formerly presented approaches, we performed recognition experiments with the Dutch version of the commercially available state-of-the-art Nuance VoCon 3200 engine.[1] The engine was a black box for us, but nevertheless it permitted us to investigate some of the proposed recipes as it was delivered with two acoustic models:

- AC-MONO: a monolingual acoustic model that was trained on native Dutch speech. The underlying phoneme set consists of 45 phonemes.
- AC-MULTI: a multilingual acoustic model that was trained on the same Dutch speech, but supplemented with equally large amounts of UK English, French and German speech. The underlying phoneme set consists of 80 phonemes and models of phonemes appearing in multiple languages have thus seen data from all these languages.

Experiments were conducted on the Autonomata Spoken Name Corpus (ASNC).[2] This corpus contains isolated proper name utterances from 240 speakers, and each speaker has read 181 names (person names and geographical names). The *speaker tongue*, defined as the mother tongue of the speaker, and the *name source*, defined as the language of origin of the name, in the ASNC is either Dutch, English, French, Turkish or Moroccan Arabic. In what follows, we have split the corpus into cells on the basis of these variables. The cell (DU,EN) for instance, contains the recordings of Dutch speakers reading English names. A division in training and test data (70–30 %) was made in such a way that any overlap between speakers and names in the two sets was avoided. In the present chapter, the training set is only used to provide phonemic transcriptions for examplary names that do not occur in the test set. No knowledge about the speech recordings, through e.g.

[1] www.nuance.com/for-business/by-product/automotive-products-services/vocon3200/index.htm

[2] For a detailed corpus description, we refer the reader to Chap. 4 of this book, Sect. 4.2, p. 62.

Table 14.1 Number of tokens per (speaker tongue, name source) combination in the ASNC test set

(DU,DU)	(DU,EN)	(DU,FR)	(DU,NN2)	(DU,ALL)
4,440	851	414	992	6,697
(DU,DU)	(EN,DU)	(FR,DU)	(NN2,DU)	(ALL,DU)
4,440	1,800	720	2,280	9,240

auditorily verified transcriptions, is employed (in contrast to [17], where we did use that information).

For the interpretation of results, a distinction was made between the native language (Dutch), non-native languages most native speakers speak/understand to some extent (English and French, called NN1 languages), and non-native languages most speakers are not familiar with at all (Turkish and Moroccan Arabic). The latter two languages are always pooled to form one 'language' called NN2. Table 14.1 shows the number of test set utterances in the different (speaker tongue, name source) cells of interest.

We chose to employ the Name Error Rate (NER) as our evaluation metric. It is defined as the percentage of name utterances that are not correctly recognised.

Figure 14.1 shows how the NER in the considered cells is affected by (1) decoding the utterances with a monolingual/multilingual acoustic model, (2) including foreign G2P transcriptions in the lexicon, and (3) adopting a monolingual/multilingual phoneme set in the lexicon.[3] The recognition vocabulary consists of all the 3,540 unique names appearing in the ASNC.

The three most important conclusions that can be drawn from the figure are the following:

1. Supplementing the lexicon with transcriptions emerging from a non-native G2P converter helps a lot for the recognition of non-native names originating from the corresponding language (the English/French transcriptions were only added for the French/English names).
2. Replacing a monolingual by a multilingual acoustic model significantly raises the recognition accuracy for non-native speakers reading native names, at least as long as the non-native language under concern was included in the acoustic model training data.
3. Nativising the non-native G2P transcriptions does not (significantly) reduce the gains that can be achieved with a multilingual acoustic model.

The first two conclusions confirm the formerly cited observations and the fact that in the target applications, the two techniques act complementary. The last conclusion

[3] A monolingual phoneme set implies that we need *nativised* Dutch versions of the foreign G2P transcriptions. These were obtained by means of a manual mapping of the foreign phonemes onto the Dutch phoneme set. The mapping was based on our own linguistic intuition, without prior knowledge of the recordings.

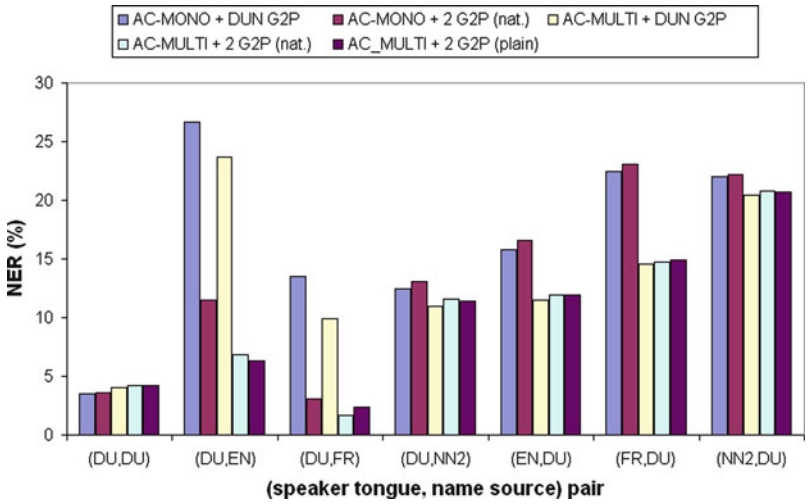


Fig. 14.1 NER results per ASNC cell for five different systems which differ in (a) the acoustic model (monolingual = AC-MONO, multilingual = AC-MULTI), (b) the G2P transcriptions included in the lexicon (DUN G2P = only a Dutch transcription, 2 G2P = additional English/French transcription for English/French names), and (c) the use of plain or natvised foreign G2P transcriptions

was published for the first time in [18]. It suggests that native speakers articulate foreign sounds with a native accent.

Based on the above conclusions we defined a state-of-the-art baseline system against which we will measure the effect of our lexical modeling approaches. Our baseline comprises a multilingual acoustic model (AC-MULTI) and a lexicon of pronunciations emerging from a Dutch, a French and an English G2P converter, in which the foreign transcriptions are natvised.

14.3 Potential for Further Improvement

The former experiments tell us what can be achieved with a lexical model based on existing general-purpose G2P converters. But what would a more advanced model be able to achieve? Imagine for instance that the lexicon contains for each name all actually used transcriptions of that name. How good would the recognition be then?

To test this situation, we supplemented the baseline lexicon with all auditorily verified transcriptions that were found in the training and test utterances of the ASNC. This resulted in a lexicon with 8.7 transcriptions per name on average. The improvements obtained with this lexicon (Table 14.2) were substantial for all cells. This makes it plausible that lexical modeling is able to yield a significant improvement over the baseline system.

Table 14.2 NER (%), per name source and per speaker tongue, for the baseline system and for a system with a lexicon that also comprises all actually used pronunciations per name

System	(DU,DU)	(DU,EN)	(DU,FR)	(DU,NN2)	(DU,ALL)
Baseline	4.2	6.8	1.7	11.6	5.5
Cheat	2.8	2.8	1.4	1.5	2.5
System	(DU,DU)	(EN,DU)	(FR,DU)	(NN2,DU)	(ALL,DU)
Baseline	4.2	11.9	14.7	20.8	10.6
Cheat	2.8	3.4	6.4	9.6	4.9

14.4 A Novel Pronunciation Modeling Approach

The proposed method creates pronunciation variants on the basis of automatically derived stochastic transformation rules that convert the phonemic output of a standard G2P into transcriptions that are more appropriate for names. Each rule predicts with which probability a phoneme sequence (called the *focus*) appearing in the initial G2P transcription (called the *source* transcription) may be phonetised as an alternative phoneme sequence (called the *rule output*) when it occurs in a particular linguistic context that can be defined in a flexible way (see below). The rules for a certain focus are embedded in the leaf nodes of a decision tree that uses yes/no-questions to distinguish between different contexts. Since the rules are stochastic in nature they will lead to multiple transcriptions per name with different probabilities. Although the VoCon engine cannot cope with these probabilities in the recognition lexicon,[4] they are still used for pronunciation selection during the lexicon creation. The presented approach constitutes a unique combination of the following features:

1. The transformable objects can be phonemic sequences (phoneme patterns) of different lengths (most published methods are confined to single phonemes).
2. The linguistic context is not restricted to the phonemic context (as in many other studies) but it can also include orthographic (graphemic), syllabic, morphological, syntactic and semantic information in a flexible way.
3. The computer-aided identification of suitable syllabic and morphological features is facilitated by built-in automatic procedures in the rule learning process.
4. The relevant (focus, output) combinations as well as the rules are learned fully automatically.

Other published methods (e.g. [12, 19–21]) share some of the above features, but we believe to be the first to propose and assess a method incorporating all these features simultaneously.

[4]This is an unfortunate limitation of the VoCon recogniser. Estimates based on preliminary experiments in which VoCon N-best hypothesis lists were rescored with the transcription variant probabilities learn that the latter can probably bring additional gains of up to 5 % relative.

The derived rules constitute a so-called P2P converter. It can be learned with the tools that were created in the first Autonomata project. All that is needed is a lexical database comprising of the order of a thousand names representative of the envisaged application domain. Per name, this database has to supply one or more plausible pronunciations and, optionally, some semantic tags (e.g. the name category). We argue that in many practical situations, such a database can be created cheaply because of its limited size, and because it can be elicited from one or two persons who are acquainted with the domain (and are able to write phonetics). These persons can select the names and enter their typical pronunciations.

Since the typical transcriptions have to be supplied by a human, the method as a whole is only semi-automatic, but once all transcriptions are available, the method is conceptually automatic. Nevertheless, it is practically implemented as a process that permits the user to intervene in an easy and transparent way if he believes that with these interventions he can surpass the improvements attainable with the automatic procedure. Note that the interventions boil down to simple updates of text files on the basis of statistical information that is being generated automatically after each step of the rule learning procedure.

Let us now review the different steps of our method, starting with a review of the contextual features we have selected.

14.4.1 Contextual Features

First of all, we consider the two phonemes immediately preceding and succeeding the focus as the primary contextual features (= 4 features). However, as in [19], we also take syllabic information into account, such as the identities of the vowels of the focus syllable and its two surrounding syllables (= 3 features) and the stress levels (no stress, primary stress or secondary stress) of these syllables (= 3 features).

Secondly, we follow the argument of Schaden [20, 21] that the orthography plays a crucial role in non-native pronunciation variation modeling because it is the key to the detection of systematic phonetisation errors. Take the French cheese name “Camembert” for instance. While the native pronunciation of this name is /“ka.ma˜.bER/, a native Dutch speaker may be inclined to pronounce it as /ka.m@m.“bErt/ because in Dutch, a “t” in the orthography is normally not deleted in the pronunciation (cf. [21] for more examples). The main limitation of Schaden’s work was that it employed handcrafted rules. In a similar vein, [12] incorporated graphemic information in an automatic data-driven approach, but the limitation of that work was that the focus had to be a single phoneme and that the graphemic context was restricted to the grapheme that gave rise to this focus. For our experiments, we considered four graphemic features: the graphemic pattern that caused the focus (but restricted to the first two graphemic units), the graphemic units immediately to the left and the right of this pattern, and a flag signaling whether or not the graphemic pattern causing the focus ends on a dot (= a simple indicator of an abbreviation).

Thirdly, we support the suggestion of Schaden [21] to consider morphological information as a potentially interesting context descriptor. Schaden noticed for instance that the vowels in the German suffixes "-stein" and "-bach" are less susceptible to accented pronunciations than the same vowels in other morphological contexts, but he did not actually build a system exploiting this observation. Since we would need multiple morphological analyzers in our cross-lingual setting, since these analyzers are expected to fail on many proper names and since we believe that a detailed morphological analysis is not very effective for our purposes, we did not try to incorporate them. Instead, we opted for a simple and pragmatic approach which automatically detects syllables, prefixes and suffixes often co-occurring with name transcription errors:

1. Three booleans indicating whether the focus syllable, the previous and the next syllable belong to a user-specified list of problematic syllables,
2. A boolean indicating whether the focus appears in a word starting with a prefix that belongs to a user-specified prefix list,
3. A boolean indicating whether the focus appears in a word ending on a suffix that belongs to a user-specified suffix list,
4. The positions (in numbers of syllables) of the focus start and end w.r.t. the first and last syllable of the name stem respectively (the name stem is obtained by depriving the name of the longest prefix and suffix from the user-specified prefix and suffix lists).[5]

Further below we will explain how to get the mentioned syllable, prefix and suffix lists in a semi-automatic way.

Finally, we believe that in the envisaged applications of proper name recognition, high-level semantic information such as the name category (e.g. street name, city name, Point of Interest), the source of the inquired name (if known), etc. are important to create more dedicated pronunciation variants. Therefore, we devised the P2P learning software so that such semantic tags can be accommodated through boolean features that are true if the tag belongs to predefined value sets (the values are character strings). In the experiments that will be discussed later, we employed the name category as a semantic feature, while the language of origin (which was supposed to be given) was used to select the proper P2P converter (e.g. the one intended for English names spoken by Dutch speakers).

14.4.2 The Overall Rule Induction Process

Since the phonemic focus patterns and the contextual features for the rule condition are not a priori known, the rule induction process is a little more complicated than usual. The process is outlined in Fig. 14.2. In general terms, the process is applied

[5] If the focus starts/ends in the selected prefix/suffix, the corresponding position is zero.

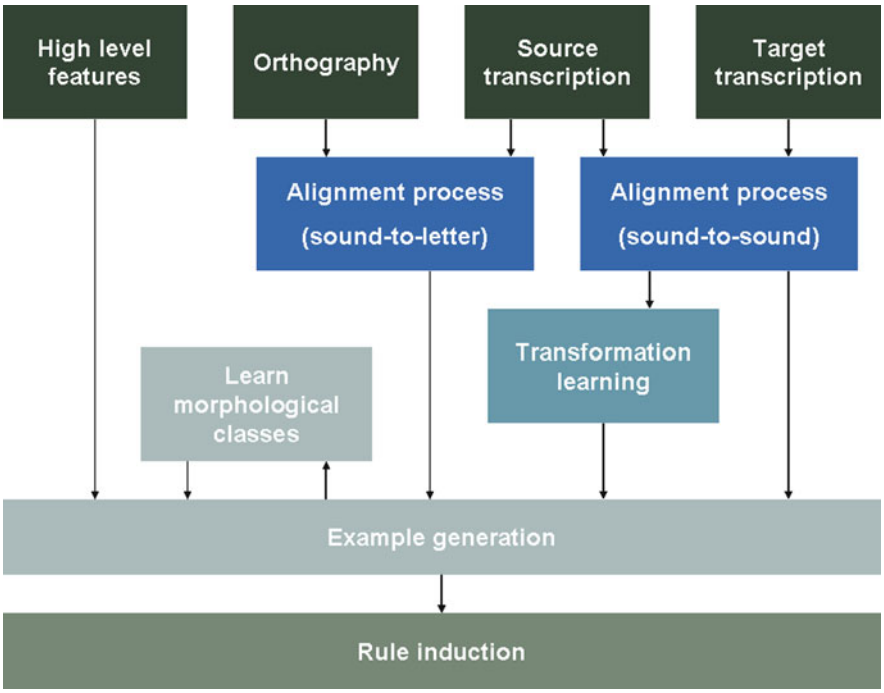


Fig. 14.2 Process for the automatic learning of a P2P converter

to a set of training objects each consisting of an orthography, a source transcription, a target transcription and a set of high-level features. Given these training objects, the learning process then proceeds as follows:

1. The objects are supplied to an alignment process incorporating two components: one for lining up the source transcription with the target transcription (sound-to-sound) and one for lining up the source transcription with the orthography (sound-to-letter). These alignments, together with the high-level features (morphological and semantic features) are stored in an alignment file.
2. The transformation learner analyzes the alignments and identifies the (focus, output) pairs that are capable of explaining a sufficiently large number of deviations between the source and the target transcriptions. These pairs are stored in a transformation file from which one can obviously retrieve the focus patterns.
3. The alignment file and the transformation file are supplied to the example generator. The latter searches for focus patterns in the source transcriptions and it generates a file containing the focus, the corresponding contextual features and the output for each detected focus pattern. These combinations will serve as the examples from which to train the rules. If no morphological features have been defined yet, one can define them on the basis of statistical information produced by the example generator. After that, one can run the example generator a second

time to create the final training examples that will also incorporate these features then.
4. The example file is finally supplied to the actual rule induction process that automatically constructs a binary decision tree per focus. Each tree is grown incrementally by choosing per leaf node the yes/no question leading to the largest entropy loss and by accepting the resulting split if this loss exceeds a predefined threshold. The rule probabilities can be derived from the counts of the different eligible outputs in each leaf node of the tree.

The full details of the approach are described in Chap. 4 of this book (cf. Sect. 4.3, p. 67) and in a journal paper [17]. We just mention here that the statistical information provided by the example generator reveals the number of co-occurrences of a discrepancy between the source and the target transcription and a syllable identity or a word property. The two word properties being considered are the graphemic sequences that correspond to the first and last one or two syllables of the word respectively. For instance, if a discrepancy frequently appears in a word starting with "vande", this "vande" will occur in the word prefix list.

14.5 Experimental Validation

In this section we investigate under which circumstances the proposed lexical methodology can enhance the name recognition performance. We first conduct experiments on the ASNC that covers the person and topographical name domains. Then, we verify whether our conclusions remain valid when we move to another domain, in casu, the POI domain.

14.5.1 Modes of Operation

Since in certain situations it is plausible to presume prior knowledge of the speaker tongue and/or the name source, three relevant modes of operation of the recogniser are considered:

- **M1**: In this mode, the speaker tongue and the source of the inquired name are a priori known. That is, the case of a tourist who uses a voice-driven GPS system to find his way in a foreign country where the names (geographical names, POI names) all originate from the language spoken in that country.
- **M2**: In this mode, the speaker tongue is known but names from different sources can be inquired. Think of the same tourist who is now traveling in a multilingual country like Belgium where the names can either be Dutch, English, French, German, or a mixture of those.

Table 14.3 NER (%), per name source and per speaker tongue, obtained with multilingual acoustic models and three distinct lexicons: (a) the baseline lexicon (2 G2P), (b) a lexicon also comprising variants generated by a P2P converter trained on the ASNC training names (ASNC), and (c) a lexicon also comprising variants generated by a P2P converter trained on an extended name set (ASNC+).

System	(DU,DU)	(DU,EN)	(DU,FR)	(DU,NN2)	(DU,ALL)
baseline (AC-MULTI + 2 G2P-nat)	4.2	6.8	1.7	11.6	5.5
baseline + 4 P2P variants (ASNC)	3.8	5.3	1.7	6.2	4.2
baseline + 4 P2P variants (ASNC+)	–	4.7	1.4	–	4.1
System	(DU,DU)	(EN,DU)	(FR,DU)	(NN2,DU)	(ALL,DU)
baseline (AC-MULTI + 2 G2P-nat)	4.2	11.9	14.7	20.8	10.6
baseline + 4 P2P variants (ASNC)	3.8	10.2	12.5	19.5	9.6

- **M3**: In this mode, neither the mother tongue of the actual user nor the source of the inquired name are a priori known. This mode applies for instance to an automatic call routing service of an international company.

The first experiments are carried out under the assumption of mode M1. In that case, we know in which cell we are and we only add variants for names that can occur in that cell. Furthermore, we can in principle use a different P2P converter in each cell. However, since for the ASNC names we only had typical native Dutch transcriptions, we could actually train only four P2P converters, one per name source. Each P2P converter is learned on a lexical database containing one entry (orthography + Dutch G2P transcription + typical Dutch transcription) per name of the targeted name source.

14.5.2 Effectiveness of P2P Variants

After having evaluated the transcription accuracy improvement as a function of the number of selected P2P variants, we came to the conclusion (cf. [17]) that it is a viable option to add only the four most likely P2P variants to the baseline lexicon. By doing so, we obtained the NERs listed in Table 14.3.

The most substantial improvement (47 % relative) is obtained for the case of Dutch speakers reading NN2 names. For the case of Dutch speakers reading French names no improvement is observed. The gains in all other cells are more modest (10–25 % relative), but nevertheless statistically significant ($p < 0.05$, even $p < 0.01$ for Dutch and NN2 names uttered by Dutch speakers.[6])

The fact that there is no gain for native speakers reading French names is partly owed to the fact that the margin for improvement was very small (the baseline 2

[6]Statistical significance of NER differences is determined using the Wilcoxon signed ranks test [22].

G2P system only makes seven errors in that cell, cf. also Table 14.2). Furthermore, the number of examples that is available for the P2P training is limited for French names. While there are 1,676 training instances for Dutch names, there are only 322 for English names, 161 for French names and 371 for NN2 names. Therefore, we performed an additional experiment in which the sets of English and French training names were extended with 684 English and 731 French names not appearing in the ASNC test set. The name set including these extensions is called ASNC+. Training on this set does lead to a performance gain for French names. Moreover, the gain for English names becomes significant at the level of $p < 0.01$ (cf. Table 14.3).

In summary, given enough typical transcriptions to train a P2P converter, our methodology yields a statistically significant ($p < 0.01$) reduction of the NER for (almost) all cells involving Dutch natives. For the utterances of non-natives the improvements are only significant at the level of $p < 0.05$ for speakers whose mother tongue is covered by the acoustic model. This is not surprising, since the Dutch typical transcriptions that we used for the P2P training were not expected to represent non-native pronunciations. Larger gains are anticipated with dedicated typical training transcriptions for these cells.

14.5.3 Analysis of Recognition Improvements

Our first hypothesis concerning the good results for native speakers was that for these speakers, there is not that much variation to model within a cell. Hence, one single TY transcription target per name might be sufficient to learn good P2P converters. To verify this hypothesis we measured, per cell, the fraction of training utterances for which the auditorily verified transcription is not included in the baseline G2P lexicon. This was the case for 33 % of the utterances in cell (DU,DU), around 50 % in (DU,EN) and (DU,FR) and around 75 % in all other cells, including (DU,NN2) for which we also observed a big improvement.

The small improvement achieved for NN2 speakers reading Dutch names is owed to the fact that many NN2 speakers have a low proficiency in Dutch reading, which implies that they often produce very a-typical phonetisations. The latter are not modeled by the Dutch typical transcriptions in our lexical database. Another observation is that NN2 speakers often hesitate a lot while uttering a native name (cf. [23]) and these hesitations are not at all modeled either.

In order to find an explanation for the good results for Dutch speakers reading NN2 names, we have compared two sets of P2P converters: one trained towards typical transcriptions and one trained towards ideal (auditorily verified) transcriptions as targets. We have recorded how many times the two P2P converters correct

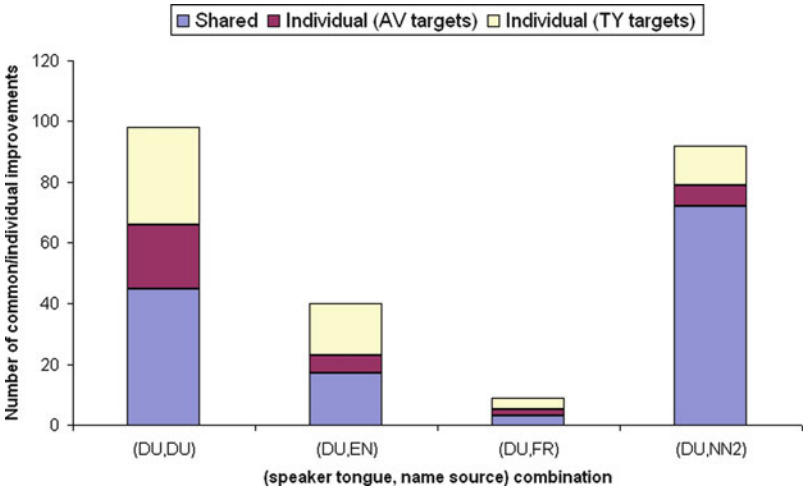


Fig. 14.3 Number of error corrections that can be achieved with the variants generated by the P2P converter trained towards typical (TY) targets, the P2P converter that was trained towards auditorily verified (AV) targets, and both P2P converters (shared)

the same recognition error in a cell and how many times only one of them does. Figure 14.3 shows the results for the four cells comprising Dutch speakers.[7]

It is remarkable that in cell (DU,NN2) the percentage of errors being corrected by both P2P converters is significantly larger than in the other cells. Digging deeper, we came to the conclusion that most of these common corrections were caused by the presence of a small number of simple vowel substitution rules that are picked up by both P2P converters as they represent really systematic discrepancies between the G2P and the typical transcriptions. The most decisive rules express that the frequently occurring letter "u" in NN2 names (e.g. Curukluk Sokagi, Butrus Benhida, Oglumus Rasuli, etc.) is often pronounced as /u/ (as in "b**oo**t") while it is transcribed as /Y/ (like in "m**u**d") or /y/ (like in the French "cr**u**") by the G2P converter.

Similarly, we have also examined for which names the P2P variants make a positive difference in the other cells. Table 14.4 gives some representative examples of names that were more often correctly recognised after we added P2P variants.

An interesting finding (Table 14.4) is that a minor change in the name transcription (one or two phoneme modifications) can make a huge difference in the recognition accuracy. The insertion of an /n/ in the pronunciation of "Duivenstraat" for instance leads to five corrected errors out of six occurrences.

[7]Note that we actually obtained these results with a system comprising a larger recognition vocabulary of 21K person and geographical names. For more details we refer to [17].

Table 14.4 Examples of proper names for which the recognition improves. Listed are: (a) the name, (b) its baseline transcription(s), (c) the P2P variant that led to an error reduction, (d) the netto number of improvements versus the number of occurrences of a name

Name	Baseline G2P variant(s)	Helping P2P variant	Netto positive result
Duivenstraat	"d9y.v@.stra:t	"d9y.v@n.stra:t	5/6
Berendrecht	b@."rEn.drExt	"be:.rEn.drExt	4/6
Carter Lane	"kAr.t@r#"la:.n@ "kA.t@#"le:jn	"kAr.t@r#"le:.n	3/6
Norfolk	nOr."fOlk "nO.f@k	"nOr.fOk	3/6
Middlesbrough	"mIt.l@z.brux "mI.d@lz.br@	"mI.d@lz.bro:	2/6
Engreux	EN."r2:ks a˜."gr2:	EN."r2:	2/6
Renée Bastin	r@."ne:#bAs."tIn r@."ne:#ba:s."te˜	rE."ne:#bAs."te˜	3/6

Table 14.5 NER results (%) for names of different sources spoken by Dutch speakers. Shown are the results for the baseline system, the best P2P system under mode M1 and the results of the P2P system under mode M2

System	(DU,DU)	(DU,EN)	(DU,FR)	(DU,NN2)	(DU,ALL)
2 G2P, mode M1	4.2	6.8	1.7	11.6	5.5
2 G2P + 4 P2P, mode M1	3.8	4.7	1.4	6.2	4.1
2 G2P + 4 P2P, mode M2	4.0	4.9	2.2	6.9	4.4

14.5.4 Effectiveness of Variants in Mode M2

So far, it was assumed that the recogniser has knowledge of the mother tongue of the user and the origin of the name that will be uttered (mode M1). In many applications, including the envisaged POI business service, a speaker of the targeted group (e.g. the Dutch speakers) can inquire for names of different origins. In that case, we can let the same P2P converters as before generate variants for the names they are designed for, and incorporate all these variants simultaneously in the lexicon. With such a lexicon we got the results listed in Table 14.5. For the pure native situation, the gain attainable under mode M2 is only 50 % of the gain that was achieved under mode M1. However, for cross-lingual cases (apart from the French names case), most of the gain achieved under mode M1 is preserved under mode M2. Note that in case of the French names, the sample size is small and the difference between 1.4 and 2.2 % is only a difference of three errors.

Table 14.6 NER results (%) for Dutch name spoken by non-native speakers. Shown are the results for the baseline system, the best P2P system under mode M1 and the results of the P2P system under mode M2

System	(DU,DU)	(EN,DU)	(FR,DU)	(NN2,DU)	(ALL,DU)
2 G2P, mode M1	4.2	11.9	14.7	20.8	10.6
2 G2P + 4 P2P, mode M1	3.8	10.2	12.5	19.5	9.6
2 G2P + 4 P2P, mode M2	4.0	10.9	13.9	20.2	10.1

14.5.5 Effectiveness of Variants in Mode M3

In case neither the mother tongue of the speaker nor the origin of the name is given beforehand (mode M3), the recognition task becomes even more challenging. Then variants for all name sources and the most relevant speaker tongues have to be added at once.

Since we had no typical non-native pronunciations of Dutch names at our disposal, a fully realistic evaluation of mode M3 was not possible. Consequently, our lexicon remained the same as that used for mode M2, meaning that the results for native speakers remain unaffected. The results for the non-native speakers are listed in Table 14.6. They are put in opposition to the baseline results and the results with lexical modeling under mode M1. The figures show that in every cell about 50 % of the gain is preserved. This implies that lexical modeling for proper name recognition in general is worthwhile to consider.

14.5.6 Evaluation of the Method in the POI Domain

In a final evaluation it was verified whether the insights acquired with person and typographical names transfer to the new domain of POI. For the training of the P2P converters we had 3,832 unique Dutch, 425 unique English and 216 unique French POI names available, each delivered with one or more plausible native Dutch transcriptions[8] and a language tag. Since there was a lot less training material for French and English names, we also compiled an extended dataset (POI+) by adding the French and English training instances of the ASNC+ dataset.

For the experimental evaluation of our method, we used the POI name corpus that was created in Autonomata Too, and that is described in Chap. 4 of this book (cf. Sect. 4.5, p. 74) and in [23].

Here we just recall that Dutch speakers were asked to read Dutch, English, French and mixed origin (Dutch-English, Dutch-French) POI, while foreign speakers were asked to read Dutch and mixed origin POI only. The recordings were

[8]The number of actual training instances per language was 6,681 for Dutch, 991 for English and 486 for French.

Table 14.7 NER results (%) for POI of different sources spoken by Dutch speakers. Shown are the results for the baseline system (2 G2P) and the P2P systems under modes M1 and M2

System	(DU,DU)	(DU,EN)	(DU,FR)	(DU,ALL)
2 G2P, mode M1	7.7	7.8	9.6	8.5
2 G2P + 4 P2P, mode M1 (POI)	6.6	6.9	8.4	7.5
2 G2P + 4 P2P, mode M1 (POI+)	–	6.9	8.1	7.3
2 G2P + 4 P2P, mode M2 (POI)	6.8	7.7	9.3	8.2
2 G2P + 4 P2P, mode M2 (POI+)	–	7.6	9.0	8.1

Table 14.8 NER results (%) for Dutch POI spoken by non-native speakers. Shown are the results for the baseline system (2 G2P) and the P2P systems under modes M1 and M2

System	(DU,DU)	(EN,DU)	(FR,DU)	(NN2,DU)	(ALL,DU)
2 G2P, mode M1	7.7	13.6	8.8	22.8	15.0
2 G2P + 4 P2P, mode M1	6.6	13.0	8.4	22.1	14.3
2 G2P + 4 P2P, mode M2	6.8	13.0	8.4	22.6	14.5

conducted such that the emphasis was on the cases of Dutch natives reading foreign POI and on non-natives reading Dutch POI.

The vocabulary of the recogniser consisted of 10K POI: all POI spoken in the POI name corpus, supplemented with additional POI that were drawn from background POI lexica provided by TeleAtlas. There was no overlap between this vocabulary and the POI set that was available for P2P training. Also, none of the POI occur in the ASNC.

Table 14.7 shows NER results for Dutch utterances under the assumptions of modes M1 and M2 respectively. Table 14.8 depicts similar results for the non-native speakers.

The data support the portability of our methodology. Adding P2P variants for POI in mode M1 strongly reduces the NER for Dutch native speakers and modestly improves the recognition for non-native speakers. In mode M2, the over-all result still holds that a substantial part of the gain is preserved. However, there are differences in the details. We now see a good preservation of the gain obtained in the purely native case, but the gains in the cross-lingual settings are more diverse. The preserved gain ranges from only 22 % (for Dutch speakers reading English names, with an extended training set) to 100 % (for English and French speakers reading Dutch names).

Furthermore, we see how an extended training set for English and French POI yields no improvement for English POI and only a small gain for French POI. This either reflects that the ASNC proper name transcriptions are not suited as training material for POI names, or that relevant information regarding the "correct" transcription of proper names can already be captured with a limited training set of name transcriptions. To verify the latter hypothesis, we performed two additional mode M1 recognition experiments for Dutch POI in which only one fourth (corresponding to about 1K unique names, 1.7K training instances) and one sixth (corresponding to about 1K training instances, for nearly 650 unique

training names) of the training set names for Dutch POI were included for the P2P converter training. We found that for both set-ups the NER was even (slightly) lower than before (6.4 % for 1K unique training POI names and 6.5 % for 1K training instances). We therefore argue that a limited training set of around 1K transcribed training names will typically be sufficient to learn a good P2P converter.

A qualitative evaluation of the improvements induced by the P2P transcriptions has been performed as well and is described in [24]. That evaluation confirmed that relatively simple phoneme conversions (substitutions, deletions, insertions) account for most of the obtained NER gains, but that a large number of more structural variations (e.g. syllable-size segment deletion) is not modeled by the P2P converters. An explicit modeling of these variations, possibly by means of other techniques, could further raise the efficiency of the POI recogniser.

14.6 Conclusions

We have proposed a novel lexical modeling methodology for the automatic recognition of proper names in a monolingual and cross-lingual setting. The method was experimentally assessed and compared against a baseline incorporating existing acoustic and lexical modeling strategies that have been applied to the same problem.

Our assessment of existing methodologies demonstrated that in a cross-lingual setting, proper name recognition can benefit a lot from a multilingual acoustic model and from transcriptions emerging from foreign G2P transcribers. We have further established that the two strategies are complementary.

The newly presented lexical modeling approach is unique in its combination of interesting properties that have never been integrated in a single system. Some of these features are: the transformation of variable length phonemic patterns from a baseline transcription, the extensive use of linguistic context at multiple levels (from phonemic to semantic), the computer-assisted identification of syllabic and morphological features, the automatic learning of context-dependent stochastic rules embedded in multiple decision trees, etc. An important feature of the method is that it does not need any labeled speech data as training material nor any expertise in automatic speech recognition. The downside is of course that the user must provide a lexical database of correspondences between a name and its typical transcription. However, since the required database is small (of the order of a thousand names), it is easy and cheap to construct.

The new method was evaluated under different modes of operation differing in the a priori knowledge one has about the mother tongue of the speaker and the language of origin of the name the speaker can inquire. When both languages are a priori known, one can achieve important reductions of the name error rate: from 10 % relative for the pure native setting, over 15 % relative for the cross-lingual settings involving a non-native language that was involved in the construction of the baseline lexicon and in the training of the multilingual acoustic models, to 45 % relative for the case where Dutch speakers read non-native names of a language

they are not familiar with. Note that the proposed method is currently not able to cope with the hesitations and strongly a-typical pronunciations of Dutch names by speakers with a low proficiency in Dutch.

References

1. Cremelie, N., ten Bosch, L.: Improving the recognition of foreign names and non-native speech by combining multiple grapheme-to-phoneme converters. In: Proceedings ISCA ITRW on Adaptation Methods for Speech Recognition, Sophia Antopolis, France, pp. 151–154 (2001)
2. Eklund, R., Lindstrom, R.: How foreign are 'foreign' speech sounds? Implication for speech recognition and speech synthesis. In: Proceedings RTO Meeting on Multi-Lingual Interoperability in Speech Technology, Hull, Canada, pp. 15–19 (1999)
3. Flege, J.: The production and perception of foreign language speech sounds. In: Winitz, H. (ed.) Human Communication and Its Disorders. A Review, pp. 224–401. Norwood, Ablex (1988)
4. Schaden, S.: Regelbasierte Modellierung fremdsprachlich akzentbehafteter Aussprachevarianten. PhD Dissertation University of Bochum (2006)
5. Trancoso, I., Viana, C., Mascarenhas, I., Teixeira, C.: On deriving rules for nativised pronunciation in navigation queries. In: Proceedings Eurospeech, Budapest, Hungary, pp. 195–198 (1999)
6. Maison, B., Chen, S., Cohen, P.: Pronunciation modeling for names of foreign origin. In: Proceedings ASRU, Virgin Islands, USA, pp. 429–434 (2003)
7. Schultz, T., Kirchhof, K.: Multilingual Speech Processing. Elsevier, Academic (2006)
8. Stouten, F., Martens, J.: Recognition of foreign names spoken by native speakers. In: Proceedings Interspeech, Antwerp, Belgium, pp. 2133–2136 (2007)
9. Leggetter, C., Woodland, P.: Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. Comput. Speech Lang **9**, 171–185 (1995)
10. Gauvain, J.-L., Lee, C.-H.: Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. IEEE Trans. Speech Audio Process. **2**, 291–298 (1994)
11. Mayfield-Tomokiyo, L., Waibel, A.: Adaptation methods for non-native speech. In: Proceedings Workshop on multilinguality in Spoken Language Processing, Aalborg, Denmark (2001)
12. Bouselmi, G., Fohr, D., Illina, I., Haton, J.: Fully automated non-native speech recognition using confusion-based acoustic model integration and graphemic constraints. In: Proceedings ICASSP, Toulouse, France, pp. 345–348 (2006)
13. Stemmer, G., Nöth, E., Niemann, H.: Acoustic modeling of foreign words in a german speech recognition system. In: Proceedings Eurospeech, Aalborg, Denmark, pp. 2745–2748 (2001)
14. Li, Y., Fung, P., Xu, P., Liu, Y.: Asymmetric acoustic modeling of mixed language speech. In: Proceedings ICASSP, Prague, Czech Republic, pp. 5004–5007 (2011)
15. Bartkova, K., Jouvet, D.: Using multilingual units for improving modeling of pronunciation variants. In: Proceedings ICASSP, Toulouse, France, pp. 1037–1040 (2006)
16. Bartkova, K., Jouvet, D.: On using units trained on foreign data for improved multiple accent speech recognition. Speech Commun. **49**(10–11), 836–846 (2007)
17. Reveil, B., Martens, J., van den Heuvel, H.: Improving proper name recognition by means of automatically learned pronunciation variants. Speech Commun. **54**(3), 321–340 (2012)
18. Réveil, B., Martens, J.-P., D'Hoore, B.: How speaker tongue and name source language affect the automatic recognition of spoken names. In: Proceedings Interspeech, Brighton, UK, pp. 2995–2998 (2009)

19. Riley, M., Byrne, W., Finke, M., Khudanpur, S., Ljolje, A., McDonough, J., Nock, H., Saraclar, M., Wooters, C., Zavaliagkos, G.: Stochastic pronunciation modelling from hand-labelled phonetic corpora. Speech Commun. **29**(2–4):209–224 (1999)
20. Schaden, S.: Rule-based lexical modelling of foreign-accented pronunciation variants. In: Proceedings 10th EACL Conference, Budapest, Hungary, pp. 159–162 (2003)
21. Schaden, S.: Generating non-native pronunciation lexicons by phonological rules. In: Proceedings ICPhS, Barcelona, Spain, pp. 2545–2548 (2003)
22. Conover, W.: Practical Nonparametric Statistics, vol. 3. Wiley, New York (1999)
23. Schraagen, M., Bloothooft, G.: Evaluating repetitions, or how to improve your multilingual asr system by doing nothing. In: Proceedings LREC, Valletta, Malta, pp. 612–617 (2010)
24. Schraagen, M., Bloothooft, G.: A qualitative evaluation of phoneme-to-phoneme technology. In: Proceedings Interspeech, Florence, Italy, pp. 2321–2324 (2011)

Chapter 15
N-Best 2008: A Benchmark Evaluation for Large Vocabulary Speech Recognition in Dutch

David A. van Leeuwen

15.1 Introduction

Automatic Speech Recognition (ASR) is a discipline of engineering that benefits particularly well from formal evaluations. There are several reasons for this. Firstly, speech recognition is basically a pattern recognition task, and to scientifically show that the system works it needs to be tested on fresh material that has never been observed by the system, or indeed the researchers themselves. This means that speech material for testing purposes needs to be collected, which requires quite some effort, but can formally only be used once. It is therefore more efficient if the evaluation material is used to determine the performance of several systems simultaneously, which suggests a common form of this kind of performance benchmarking: that of a formal evaluation. Secondly, after a system evaluation the evaluation material and protocol can be used for future researchers as a benchmark test: algorithms can be developed and tuned to increase performance on the test. By using a well-established formal evaluation protocol performance figures can directly be compared amongst different researchers in the literature, which gives more meaning to the actual figures. Thirdly, a benchmark evaluation gives researchers a clear focus and goal, and appears to stimulate the different research groups to get the best out of their system in a friendly competitive way.

Formal evaluations in speech technology have their origin in the early 1990s of the last century, when the US Advanced Research Projects Agency (ARPA) organised regular evaluations in speech recognition executed by the National Institute of Standards and Technology (NIST) [16], soon followed by speaker [12] and language [13] recognition. In the early years the language of interest for speech recognition invariably was English, but as tasks got harder and performance got

D.A. van Leeuwen (✉)
Centre for Language and Speech Technology, Nijmegen, The Netherlands
e-mail: d.vanleeuwen@let.ru.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_15, © The Author(s) 2013

better, also Arabic and Mandarin became target languages. The NIST evaluation campaigns were so successful that researchers in Europe followed the good example of the US and held their own evaluations of speech technology. One such evaluation was the EU-funded project SQALE[1] [24], in which large vocabulary speech recognition systems (20–65k words) were tested in British and American English, French and German, using read speech. Later, the French *Technolangue* program encompassed *Evalda*, the evaluation of many different human language technologies, among which the ESTER[2] evaluation for Broadcast News speech.

The idea of evaluating technology regularly is the so-called *evaluation paradigm* where system performance is driven to improve over time because researchers compare their approaches in the previous evaluation, gather the best ingredients and implement this in their systems for the next evaluation round. In speech, this paradigm has been implemented most clearly by NIST campaigns, the Technolangue program and Evalita.[3] Other efforts in evaluation, e.g., SQALE and the NFI-TNO Forensic Speaker Recognition Evaluation [23], do no re-occur, and therefore unfortunately do not have the same effect on system performance.

Needless to say, the speech recognition systems require vast amounts of training resources, such as annotated speech material for acoustic models and large quantities of textual material for building language models. These resources were collected and very effectively shared with the research community through the Linguistic Data Consortium (LDC), which again found its European counterpart in the European Language Resources Association (ELRA). In 1998 the Dutch Language Union started a project *Corpus Gesproken Nederlands* (CGN, Spoken Dutch Corpus [14]) aiming at collecting about ten million words of speech as it was spoken by adults in The Netherlands and Flanders at the time. The CGN was created for general linguistic research, and not specifically for the development of a specific speech technology. It thus encompassed many different speech styles, but some of these were indeed suitable for building speech recognition systems for the typical speech recognition task at that time.

Around 2005 there were several research institutions in the low countries that had developed speech recognition systems for the Dutch language [2, 15]. Some were using CGN [20], others used their own databases [10, 15]. The different data used for evaluation and training made it difficult to value the merits of the various systems used. In The Netherlands and Flanders we seemed to be in a situation where there was technology and training material available, but no official speech recognition benchmark evaluation to compare these systems. The STEVIN project *N-Best* aimed at setting up the infrastructure for conducting a benchmark test for large vocabulary ASR in the Dutch language, and collecting data, performing the evaluation and disseminating the results and evaluation data. The acronym N-Best originally is of Dutch origin (*Nederlandse Benchmark Evaluatie voor SpraakTechnologie*) but also

[1] Speech recognition Quality Assessment for Linguistic Engineering

[2] Evaluation des Systèmes de Transcription Enrichie d'émissions Radiophoniques

[3] Evaluation of Natural Language Processing and Speech Tools for Italian, www.evalita.it

has the English interpretation Northern and Southern Dutch Benchmark Evaluation for Speech Technology,[4] expressing the somewhat political wording necessary to indicate the two major language variations in Dutch commonly known as Dutch and Flemish.

This chapter is organised as follows. In Sect. 15.2 the N-Best project is reviewed, then in Sect. 15.3 the evaluation protocol is described. Then, in Sect. 15.4 the evaluation results are presented and discussed.

15.2 The N-Best Project

The project N-Best was funded by the Dutch Language Union research programme STEVIN and consisted of seven partners in three different roles. The coordinator was TNO,[5] responsible of actually carrying out the evaluation. The Nijmegen organisation SPEX[6] was responsible for recording and annotating the evaluation data, and five partners from universities in The Netherlands and Flanders were contributing by developing speech recognition systems for the specific tasks in N-Best and processing the evaluation material. These were ELIS[7] from the Ghent University, ESAT[8] from the Katholieke Universiteit Leuven, the CLST[9] from Radboud University Nijmegen, EWI[10] from the Delft University of Technology, and HMI[11] from the University of Twente.

Despite the competitive nature that a formal evaluation has, N-Best was a collaborative project. In several of the steps that needed to be taken all partners, including the ones with systems under evaluation, collaborated in order to make it feasible for the partners with less experience in evaluation or even large vocabulary speech recognition for Dutch. Most notably, ESAT provided the necessary relation with Mediargus, the supplier for Southern Dutch news paper texts for language model training, and HMI did likewise with their relation with the publisher PCM, the supplier of Northern Dutch newspaper data. Some text-normalising code was shared between partners, and in some cases an entire language model was shared.

[4] Obviously, the term 'Technology' is too broad for a project only dealing with ASR, but this term makes the acronym nicer. Moreover, it can serve as an umbrella name for possible future speech technology evaluations in the low countries.

[5] Netherlands Organisation for Applied Scientific Research

[6] Speech Processing Expertese centre

[7] Electronics and Information Systems department

[8] Department of Electrotechnical Engineering

[9] Centre for Language and Speech Technology

[10] Faculty Electrical Engineering, Mathematics and Computer Science

[11] Human Media Interaction

15.2.1 Specification of the Task and Evaluation Protocol

One of the first things that needed to be established was the definition of the evaluation protocol. Although this was primarily a task of the coordinator, preliminary versions of the document were discussed among all project partners and omissions or errors were pointed out. The result of this process was the publication of the 2008 N-Best evaluation plan [21]. The evaluation plan was inspired by several similar documents from NIST and from the ESTER project, and adapted for the task that was defined for N-Best. The main task was the transcription of Dutch speech, both in the Northern and Southern Dutch variety, and in both the speech styles "Broadcast News" (BN) and "Conversational Telephone Speech" (CTS), amounting to four 'primary tasks'. These styles were well known in the speech recognition community, and well studied in the case of (American) English. Further, the main training condition was to use a specified partition of CGN for acoustical training, and newspaper text provided by partners ESAT and HMI.

15.2.2 Recruitment of Participants

One of the objectives of the N-Best projects was to establish the state-of-the-art of automatic speech recognition for Dutch. In order for this level of performance to be representative of what current technology was capable of, it was important that several of the best laboratory systems take part in the evaluation. Therefore one of the tasks in the N-Best project was to find sites that were willing to participate in N-Best without direct funding from the project. Given the fact that there are not many speakers of Dutch in the world, and that the development of a speech recognition system for a new language requires quite some effort, it was not trivial to find researchers outside the low countries that would participate in the evaluation. Still, we found two teams in Europe that registered: the combination Vecsys[12] Research + Limsi from Paris, France and Brno University of Technology from Brno, Czech Republic. One site registered with the idea of testing a commercial speech recognition system, but had to pull out because the task was too hard.

15.2.3 Testing of the Infrastructure

Because for most ASR partners in the project this was their first formal evaluation, and for TNO it had been over a decade since it had been involved in a speech recognition evaluation, it was decided to have a dry-run in order to test the evaluation

[12]Now Vocopia

process and protocol with respect to file formats, recognition output, file exchange, and scoring. In order to carry this out, some test material was necessary, and for this we utilised parts of the acoustic training material that were marked for development testing. In order to simulate the typical train-test data shift as well as possible within the larger collection of the CGN, the development test data was selected based on recording date. Because the recording date was not available for all parts in the CGN in the standard release from the Dutch *HLT Agency*, a special contract was signed between the coordinator and the HLT Agency, so that the coordinator was able to split off development test material from the training data based on the actual recording date.

Most of the N-Best project partners submitted results for this development test material, and this was scored by the coordinator, such that submission formats and scoring scripts could be tested. The experiences were discussed in an N-Best project workshop. The result was that some of the writing conventions were clarified in the evaluation plan, and that scoring scripts were improved. The development test material, including scoring scripts and the scores of one of the partners, was distributed amongst all N-Best evaluation participants.

15.2.4 Recording, Annotating, and Selection of the Evaluation Data

The evaluation data was recorded by partner SPEX. For the Broadcast News (BN) speech data, material was obtained digitally from the copyright holders, with whom license agreements were set up such that the material could be used for this evaluation, and could further be distributed by the Dutch Language Union. For Conversational Telephone Speech (CTS) data subjects were recruited from a variety of locations within Flanders and The Netherlands. The recruitment strategies allowed for partners in telephone conversations to be familiar with each other – this typically leads to more spontaneous speech which makes it a harder transcription task. In order to stimulate the conversation, subjects were given a topic to discuss from a predefined list of topics, similar to how Switchboard [7] was set up. However, the subjects were free to deviate from this topic. The level of familiarity between subjects and actual topic were not explicitly annotated.

About 3 h of speech for each primary task were recorded. These were all orthographically annotated, using a protocol very similar to the one used in the production of the CGN [8]. This data was sent to the coordinator, who made a further selection in this data, based on criteria such as the speaker's sex and regional variety for CTS, and removing ads and non-Dutch speech from the BN material. After the selection there remained a little over 2 h for each of the four tasks. This selection was then verified by SPEX, with a different transcriber than in the first annotation round. Finally the coordinator listened to all speech prior to sending the data to the participants, and manually remove the last glitches in the data.

15.3 The N-Best Evaluation

The N-Best evaluation was held in April 2008. The evaluation protocol was described in the Evaluation Plan document [21]. The main characteristics of the evaluation protocol are reviewed in this section.

15.3.1 Task

The task in N-Best is that of automatic *transcription* of speech. The speech material is conditioned to one of four domains. The regional variants of Dutch are Northern and Southern Dutch (also known as *Dutch* and *Flemish*). The speech material is obtained from the either radio and television shows (Broadcast News, BN) or telephone conversations (Conversational Telephone Speech, CTS), which are referred to as speech styles. The four primary tasks in N-Best are to automatically transcribe 2 h of speech in each of the four domains formed by the Cartesian product of regional variant and speech style.

15.3.2 Conditions

Several conditions are defined under which the ASR systems should operate. One set of conditions are known as the primary conditions. All participants must submit recognition hypothesis results for each of the primary tasks in the primary condition. Further, sites are encouraged to submit results of any of the task in contrastive operating conditions, where a set of predefined contrastive conditions are suggested. Other important resources for a recognition system, such as pronunciation dictionary, were considered part of the system design and were not controlled or restricted.

15.3.2.1 Primary Conditions

Training material

In the primary condition the training material for acoustic and language models was limited to the material designated and distributed within the N-Best evaluation. The acoustic training material consisted of designated parts of the CGN, as shown in Table 15.1. The language model training material consisted of newspaper text, as distributed by the coordinator. This material was contributed by two of the N-Best partners, also participants, to the evaluation. All language model training material originated from before 1 January 2007, which was the limit for language model training material in *any* of the conditions. The specification of written sources is found in Table 15.2.

Table 15.1 Specification of the acoustic training components of CGN

		Duration (h)	
Speech domain	Component	Northern	Southern
Broadcast news	f: broadcast interviews	42.9	20.9
	i: live commentaries	30.7	12.9
	j: news/reports	8.3	9.1
	k: broadcast news	27.5	8.2
	l: broadcast commentaries	7.9	6.8
	Total	99.4	52.9
Conversational telephone speech	c: switchboard	55.3	36.5
	d: local minidisc	36.7	27.5
	Total	92.0	64.0

Table 15.2 Language modeling training resources for N-Best

Supplier	Newspapers	Years	Size (million words)
PCM	Algemeen Dagblad	2001–2004	66
	Dortsch Dagblad	1999–2000	1.9
	HP de Tijd	1999–2000	0.9
	NRC Handelsblad	1999–2004	82
	Het Parool	1999–2004	57
	Trouw	1999–2004	55
	Vrij Nederland	1999–2000	1.2
	De Volkskrant	1999–2004	94
	Total NL	1999–2004	360
Mediargus	De Morgen	1999–2004	135
	De Standaard	1999–2004	118
	De Tijd	1999–2004	98
	Gazet van Antwerpen	1999–2004	240
	Het Balang van Limburg	1999–2004	106
	Het Laatste Nieuws	1999–2004	284
	Het Nieuwsblad	1999–2004	322
	Het Volk	2000–2004	133
	Total VL	1999–2004	1,436

Processing Time

The primary condition for processing speed was *unlimited time*, with the condition that results needed to be submitted within the deadline, which was 25 days after the data became available. There was no restriction to the number of CPUs or cores that are used to process the data.

15.3.3 Contrastive Conditions

Training Material

Contrastive training conditions could be formed by using any acoustic or language modeling training material, as long as the material originated from before 1 Jan 2007. This is because the evaluation test material was obtained from recordings that were made after this date, and thus we could be reasonably sure that the evaluation material (in speech or text form) did not occur in any training material used.

Processing Time

Contrastive conditions in processing speed could include any speed restriction. In line with other international evaluations, we suggested the specific processing time restrictions of $1\times$ RT (real time) and $10\times$ RT.

15.3.4 Contrastive Systems

Each site was to submit primary task results for at least one system, the primary system. Participants were encouraged to submit results for other, contrastive systems, for any of the tasks in any of the conditions, as long as also primary system results were submitted for these conditions.

15.3.5 Evaluation Measure

The primary evaluation measure of performance was the Word Error Rate (WER), as calculated by NIST `sclite` tools [6].[13] In the determination of the WER non-lexical events (coughs, filled pauses, etc.) were not included in the reference transcription. However, an ASR system would have to indicate these non-lexical events as such if it recognised these events, or these would be counted as insertions.

The evaluation plan [21] specified the way numbers, compound words, acronyms, capitalisation, abbreviation, accents, punctuation should be used in the system's output. Further, relaxed interpretation of spelling was adhered to because of the many spelling reforms the Netherlands and Flanders have experienced in the past.

[13] Available from http://www.itl.nist.gov/iad/mig/tools/

15.3.6 File Formats

The files used in the evaluation were all in standard formats. Audio was distributed in RIFF/WAV files with 8 kHz A-law two-channel encoding for CTS and 16 kHz 16-bit linear PCM encoding for BN. Evaluation control files, specifying which parts of the audio were under evaluation, were in NIST Unpartitioned Evaluation Map (UEM) format [4]. The recognition hypothesis results were expected in UTF-8 encoded CTM files [4]. Specifically, only words in field 7 of type `lex` are considered in computing the WER, other types are ignored.

15.3.7 Evaluation and Adjudication

The speech data were released as a downloadable archive file containing all speech files, together with the corresponding UEM files. Results were due at the coordinator within 25 days. Results arriving late were marked as such. Within a week the coordinator released the first scoring results, including references and alignments, after which there was a 2-week adjudication period. Here, participants could question certain decisions in the scoring. Finally, the coordinator would release the final results.

15.4 Results

15.4.1 Submission Statistics

During the preparations of the evaluation [21], it was decided that in written publications comparative results [22] are to be presented anonymously, but that individual sites can of course present their own results [1,3,9]. This was inspired by the way it goes in the very successful NIST Speaker Recognition campaigns, and the most important reason for N-Best was to make the evaluation more attractive for industrial participants. However, one industrial subscription to the evaluation pulled out at the last moment, so the anonymity in this publication only serves to adhere to original agreements.

There were seven sites participating in the evaluation, including the five ASR sites from the N-Best project. Six of these submitted results before the deadline, totaling 52 submissions distributed over the four primary tasks. Each of the six sites included their primary system in these submissions. One participant (“sys 1”) refrained from receiving the first results until about 3 days after these had been sent to the other five participants, in order to finish two ‘unlimited time’ contrastive runs for their CTS system.

One of the participants (“sys 4”) did not submit results, but refrained from interaction with any of the involved parties, until about 4 months after the official

Table 15.3 Overall results of N-Best 2008. Figures indicate the WER, in %. Systems with * indicate late submissions

	bn nl	bn vl	cts nl	cts vl	Average
sys 1	17.8	15.9	35.1	46.1	28.7
sys 2	30.8	26.5	58.3	62.3	44.5
sys 3	39.3	33.5	60.9	71.5	51.3
sys 4*	41.4	25.6	75.3	69.9	53.0
sys 5	42.9	28.1	73.6	68.0	53.1
sys 6	46.5	51.5	59.3	78.7	59.0
sys 7	59.8	63.7	88.6	90.2	75.6

deadline, due to unavailability of personnel. This amount of delay is quite unusual in formal evaluations, and it is difficult to guarantee that no information about the evaluation will have reached this participant.

"Sys 3" ran three different ASR systems and four different runs of its main system. "Sys 2" ran a single-pass system contrasting its multi-pass primary system, and "sys 5" ran a contrastive language model system. Finally, 'sys 6' and 'sys 7' only submitted the required minimum of four primary tasks.

15.4.2 Primary Evaluation Results

Results for all seven primary systems in the primary conditions in all four primary tasks are shown in Table 15.3 and are plotted in Fig. 15.1. The systems are numbered in order of the average word error rate for the primary tasks. It should perhaps be noted here that 'sys 1,' showing the lowest word error rates for all tasks, submitted a $10\times$ RT system as primary system results, and had a slightly better performing 'unlimited time' contrastive system, which still is according to the rules.

We can observe from the results that CTS gives higher error rates than BN, which is consistent with results reported for English [5] . Apart from the smaller bandwidth of the audio channel, CTS also contains more spontaneous speech than the more prepared speech style that is characteristic of BN. The acoustics of CTS will also contain more regional variability compared to speech available on radio and television, so therefore the acoustic models have less spectral information to model more widely varying acoustic realisations of the sounds in the language. Another effect that makes BN data have less errors than CTS data is that the majority of the language model training material will match the linguistic content of the BN speech better than that of CTS.

15.4.3 Focus Conditions for Broadcast News Data

NIST has defined standard 'focus conditions' for the various types of speech that may appear in BN material: clean, spontaneous, and telephone speech, speech with

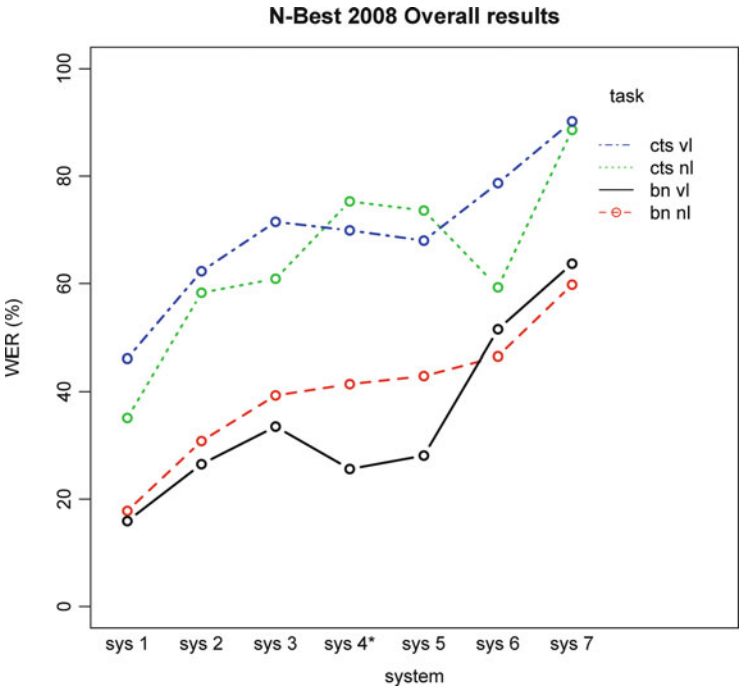


Fig. 15.1 Overall results of N-Best 2008, WER as a function of system and primary task condition. Systems are ordered according to average WER over tasks, lines connecting points are just guides for the eye. Systems with * indicate late submissions

background noise, and degraded speech. SPEX has annotated the test material for these five standard focus conditions, but in the selection criteria for the final evaluation material these conditions were not included. Hence, the amounts of data found in each of the focus conditions is not homogeneously distributed. In Table 15.4 and Fig. 15.2 the WER performance conditioned on focus condition, regional variety and speaker's sex are shown in various combinations.

Even though the performance varies widely over the different systems, ranging 10–60 %, the clean focus condition clearly has lower WER, which is not surprising. Some systems took a particularly big hit with telephone speech in the NL regional variant. This may be resulting from the way the BN training (and therefore, dry run test material) is organised in CGN: contrary to the VL variant, CGN does not contain whole news shows for the NL variant. It is conjectured that the systems that proved particularly vulnerable to telephone speech have been concentrating more on the NL part during development, and may have missed the fact that BN shows may contain this type of speech. This is consistent with the type of errors seen most for these systems in the telephone condition, deletions.

Table 15.4 BN performance expressed in WER (in %), as plotted in Fig. 15.2, but separated for Northern (*left*) and Southern (*right*) regional variants. Also indicated is the number of words N_w over which the statistics are calculated ('k' means 1,000). Systems with * indicate late submissions. Focus conditions are: all, clean speech, spontaneous speech, telephone speech, speech with background noise and degraded speech

NL	All	Clean	Spont	Tel	Back	Degr
sys 1	17.8	11.6	20.2	20.8	14.8	20.9
sys 2	30.8	23.3	33.4	37.0	25.4	32.6
sys 3	39.3	26.2	40.3	62.4	28.5	39.2
sys 4*	41.2	25.9	45.8	57.5	33.0	42.5
sys 5	42.9	27.1	49.0	58.0	33.2	41.4
sys 6	46.5	34.8	49.9	61.4	41.9	44.2
sys 7	59.8	51.0	64.8	66.4	53.4	56.3
N_w	24k4	7k2	10k2	3k8	358	2k9

VL	All	Clean	Spont	Tel	Back	Degr
sys 1	15.9	8.5	16.6	12.5	17.5	18.5
sys 2	26.5	16.6	27.6	17.8	28.1	30.4
sys 3	33.5	18.1	35.0	45.9	33.3	35.2
sys 4*	25.6	13.6	26.5	27.4	27.2	29.4
sys 5	28.1	16.4	29.5	30.1	29.2	30.1
sys 6	51.5	38.8	52.0	56.8	59.4	54.9
sys 7	63.7	59.1	61.4	57.5	72.2	73.4
N_w	22k5	2k6	13k7	873	869	4k4

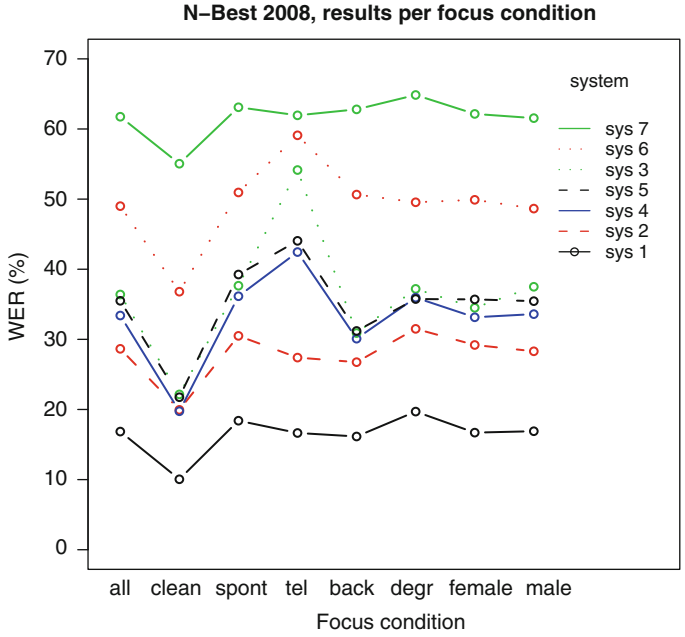


Fig. 15.2 Word error rates for each primary BN submission, analyzed over NIST focus conditions (see Table 15.4 for the legend), and separately, speaker's sex. For clarity, WERs are averaged for NL and VL accent task conditions

However, the performance of telephone speech in BN still is a lot better than in the CTS task for all systems, with notably one exception: that of 'sys 6' for NL. This systems CTS performance is actually better than in the BN telephone focus conditions. This could be explained by 'sys 6' not detecting telephone speech in NL BN data, thus not benefiting from their relatively good CTS NL acoustic models.

15.4.4 Other Accent Effect

Related to this is the analysis of results by origin of the partner. From the N-Best project partners, the partners located in Belgium performed relatively well on Southern Dutch, while the Dutch university performed better on the Northern Dutch variant. This can be appreciated from Fig. 15.3, where the interaction between the participant's home country (North for The Netherlands, South for Belgium) and regional variant of the speech is shown. This is, in a way, similar to the famous 'Other Race Effect' of human face recognition,[14] that is also observed by automatic face recognition systems [18]. We therefore coin this the 'Other Accent Effect.' We have no direct evidence why this is the case, but one reason could be the choice of phone set, the pronunciation dictionary and grapheme-phoneme conversion tools. This is one part of the ASR systems that was not specified as part of the primary training conditions. We can surmise that the researchers had better quality dictionary for their own regional accent than for the other region.

15.4.5 Contrastive System Conditions

Three sites submitted contrasting focus conditions. 'Sys 1' submitted contrasting results showing the effect of processing speed. In Fig. 15.4 it can be seen that faster processing restrictions have a negative effect on performance, but that there probably is hardly any benefit of going beyond 10× RT.

'Sys 2' ran a single-pass system as contrastive to its multi-pass primary system. The results show a quite consistent gain in WER of approximately 10 %-point for all primary tasks when running the multi-pass system (Fig. 15.5).

Finally, 'sys 3' submitted many different contrastive conditions. The main variation was in system architecture, where this site submitted results based on Soft-Sound's 'Abbot' hybrid Neural Net/Hidden Markov Model (HMM) system, [19] the site's own'SHoUT' recogniser [9] and 'Sonic' (University of Colorado, [17]) both pure HMM systems. Using SHoUT, both single and double pass system results were submitted, and additionally a 'bugfix' version of these two were scored by the coordinator. A plot comparing all of these submissions from 'sys 3' is shown in Fig. 15.6. The multi-pass systems did not improve either of the HMM systems very much, about 1 %-point for BN in both accent regions in the case of SHoUT's SMAPLR (structured Maximum A Posteriori Linear Regression) adaptation technique, and about 0.5 %-point for Sonics's CMLLR (Constrained Maximum Likelihood Linear Regression) implementation.

[14] Popularly speaking, the fact that Europeans find it difficult to recognise individual Asians, and vice versa.

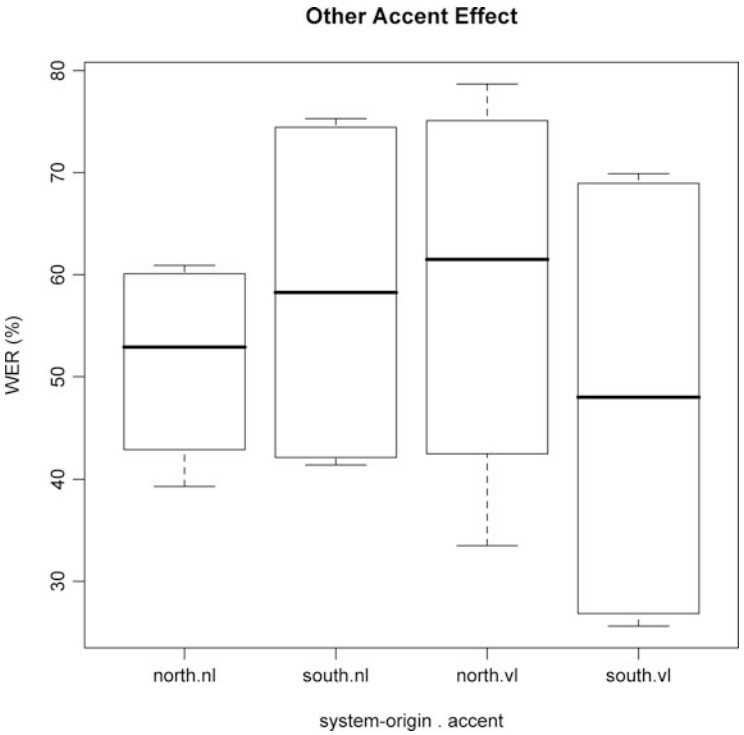


Fig. 15.3 Interaction box plot between the country of origin of the speech recognition system (North/South) and accent of Dutch (Northern – NL/Southern – VL). One system has been left out of the analysis due to extremely high error rates

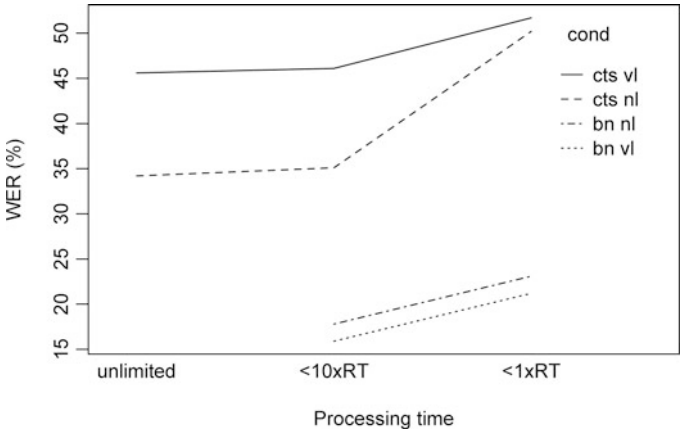


Fig. 15.4 The effect of processing speed restrictions for sys 1

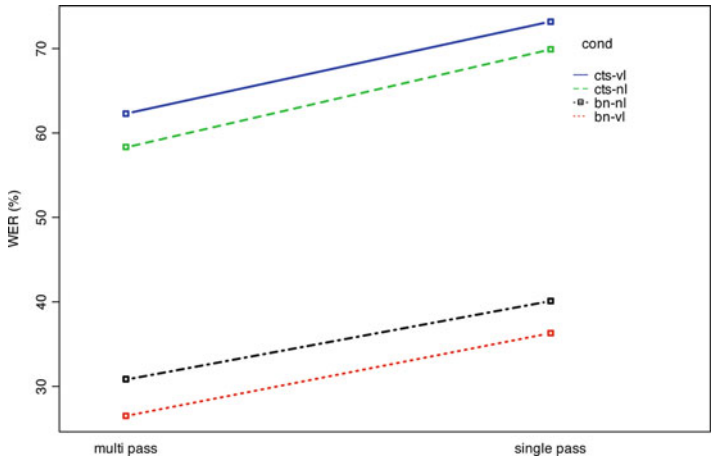


Fig. 15.5 The effect of multiple passes vs. a single pass for sys 2

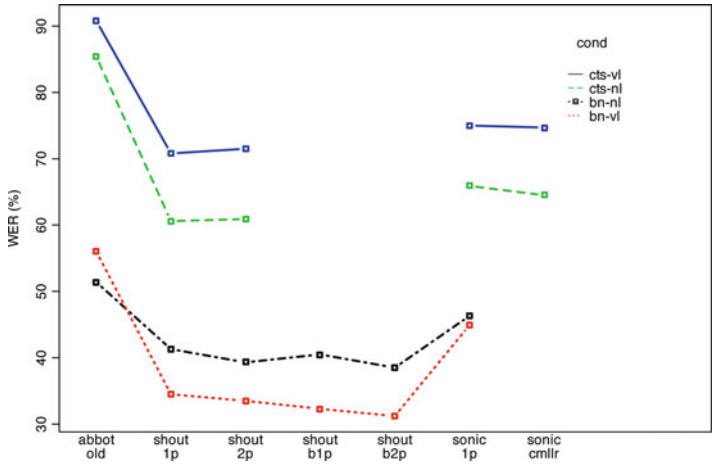


Fig. 15.6 Results for the various submissions of 'sys 3'. The primary system was 'shout 2p'

15.5 Discussion and Conclusions

From the presentations and discussions in the workshop that concluded the evaluation, it became clear that large vocabulary speech recognition in Dutch can be approached quite successfully with technology developed for languages like English, French and German. Dutch is a morphological language with strong compounding (similar to German) and is moderately inflectional. This requires large vocabularies [1, 3] of 300–500k words, but this is not uncommon for languages like German. ESAT reported language models based on morphological analysis [1], which resulted in a moderate reduction in out-of-vocabulary rate and WER during

development for smaller vocabulary sizes. Vecsys Research + Limsi [3] used a common pronunciation dictionary for Northern and Southern Dutch, which was further adapted to the task by including pronunciation frequencies obtained from forced-aligned training data. Further, most sites reported substantial efforts in text-normalisation. For instance, UTwente reported a drop in word error rate from 38.6 to 34.9 % by processing filled pauses, compounds, capitalisation, and numbers [9]. Obviously the rules in the evaluation protocol and the peculiarities of Dutch writing conventions were different enough from other languages to draw considerable efforts from the developers, but did not require radically new approaches to text normalisation. The newspaper text data distributed was from before 2005, while the date limit for contrastive conditions was 1 January 2007. However, no contrastive systems were submitted with more recent language modeling data than 2004.

The results shown in Sect. 15.4 are the outcome of first structural and comparative study of large vocabulary speech recognition for the Dutch language. The main effects observed (difference between BN and CTS, focus condition, number of passes, processing time restrictions) are consistent with what is observed in literature for speech recognition in other languages. The absolute values for the WER of the best performing systems are quite higher than for English, where very low error rates for BN are reported, of the order of magnitude of human transcription errors, and where CTS results have been reported around 15 %. The reason for this probably lies in the size of the training data, which is much smaller within N-Best than for English, where thousands of hours of acoustic training data are available. The fact that nobody submitted a contrastive system with more acoustic training data suggests that this material is not readily available to the researchers. Another reason for the higher error rates for Dutch is the fact that N-Best was the first evaluation, and that Dutch participants were not very experienced in formal evaluations, while the non-Dutch participants were not very experienced in the Dutch language, if at all. From informal inspection of the results we can conclude that the latter factor may be less important than the former.

We would like to note that in inspecting the alignments of hypothesised results with the reference transcriptions, the best performing system ‘sys 1’ caused us to notice several mistakes in the reference transcription where grammatical spelling rules or compound words were involved. We found this quite remarkable. At the same time, the scoring process details and the adjudication issues brought several difficult grammatical construction variants to the surface. Examples are *er aan* vs. *eraan*, *te veel* vs. *teveel* and *ervan uitgaan* vs. *er vanuit gaan*. The different compounding solutions in Dutch are quite hard to choose from, even for a native Dutch scholar. Although we were very lenient in the scoring process towards these issues, and spent a lot of time painstakingly checking every hypothesised error, the effect on the total WER typically was only 0.2 %-point.

Interesting may be the ‘Other Accent Effect’ observed in within the N-Best partners, that the performance for the task in their own regional language variant were better, relatively, than in the other variant. This subtle manifestation of a preference for ones own accent, even through ones own system performance, can

be compared to the 'Other Race Effect' for automatic face recognition fusion algorithms [18].

Concluding, the N-Best project can be said successful in setting up the infrastructure for a benchmark evaluation for Dutch ASR systems. The evaluation data and scoring script can be obtained from the Dutch Language Union through its data distribution agency, the HLT Agency. This includes the scores and recognition hypothesis files of the best scoring system, allowing future researchers to compare the output of their own recognition systems for Dutch to the state-of the art of 2008. The evaluation has generated at least five papers in conference proceedings [1, 3, 9, 11, 22]. It remains to be seen if follow-on evaluations of Dutch speech recognition sparks enough enthusiasm from sponsors and participating developers to be realised. This would change the N-Best evaluation from a once-off benchmark evaluation of the state of the art of Dutch ASR in 2008 to a campaign fitting in the 'evaluation paradigm' with all the benefits of exchange of knowledge and the drive to better performing speech recognition systems.

References

1. Demuynck, K., Puurula, A., Van Compernolle, D., Wambacq, P.: The ESAT 2008 system for N-Best Dutch speech recognition benchmark. In: Proceedings of ASRU, Merano, pp. 339–344 (2009)
2. Demuynck, K., Duchateau, J., Van Compernolle, D., Wambacq, P.: An efficient search space representation for large vocabulary continuous speech recognition. Speech Commun. **30**(1):37–53 (2000)
3. Despres, J., Fousek, P., Gauvain, J.-L., Gay, S., Josse, Y., Lamel, L., Messaoudi, A.: Modeling Northern and Southern varieties of Dutch for STT. In: Proceedings of Interspeech, Brighton, pp. 96–99. ISCA (2009)
4. Fiscus, J.: The rich transcription 2006 spring meeting recognition evaluation. http://www.nist.gov/speech/tests/rt/rt2006/spring/docs/rt06s-meeting-eval-plan-V2.pdf (2006)
5. Fiscus, J.G., Ajot, J., Garofolo, J.S.: The rich transcription 2007 meeting recognition evaluation. In: The Joint Proceedings of the CLEAR 2007 and RT 2007 Evaluation Workshops. Volume 4625 of LNCS, Baltimore, pp. 373–389, Springer (2007)
6. Fiscus, J.G., Ajot, J., Radde, N., Laprun, C.: Multiple dimension levenshtein edit distance calculations for evaluating automatic speech recognition systems during simultaneous speech. In: Proceedings LREC, Genoa, pp. 803–808. ELRA (2006)
7. Godfrey, J.J., Holliman, E.C., McDaniel, J.: Switchboard: telephone speech corpus for research and development. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), San Francisco, pp. 517–520 (1992)
8. Goedertier, W., Goddijn, S., Martens, J.-P.: Orthographic transcription of the Spoken Dutch Corpus. In: Proceedings of the LREC, Athens, pp. 909–914 (2000)
9. Huijbregts, M., Ordelman, R., Werff, L., Jong, F.M.G.: SHoUT, the University of Twente submission to the N-Best 2008 speech recognition evaluation for Dutch. In: Proceedings of Interspeech, Brighton, pp. 2575–2578. ISCA (2009)

10. Huijbregts, M.A.H., Ordelman, R.J.F., de Jong, F.M.G.: A spoken document retrieval application in the oral history domain. In: Proceedings of 10th International Conference Speech and Computer, Patras, pp. 699–702. University of Patras (2005)
11. Kessens, J., van Leeuwen, D.: N-Best: the Northern and southern dutch Benchmark Evaluation of Speech recognition Technology. In: Proceedings Interspeech, pp. 1354–1357, Antwerp, August 2007. ISCA.
12. Martin, A.F., Greenberg, C.S.: The NIST 2010 speaker recognition evaluation. In: Proceedings of Interspeech, Makuhari, pp. 2726–2729. ISCA (2010)
13. Martin, A.F., Le, A.N.: NIST 2007 language recognition evaluation. In: Proceedings of Speaker and Language Odyssey, Stellenbosch, South Afrika. IEEE (2008)
14. Oostdijk, N.H.J., Broeder, D.: The Spoken Dutch Corpus and its exploitation environment. In: Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03), Budapest (2003)
15. Ordelman, R.: Dutch speech recognition in multimedia information retrieval. PhD thesis, University of Twente (2003)
16. Pallett, D.: A look at NIST's benchmark ASR tests: Past, present, and future. http://www.nist.gov/speech/history/ (2003)
17. Pellom, B.: Sonic: the university of colorado continuous speech recognizer. Technical Report TR-CSLR-2001-01, University of Colorado, Boulder, March 2001
18. Phillips, P.J., Narvekar, A., Jiang, F., O'Toole, A.J.: An other-race effect for face recognition algorithms. ACM Trans. Appl. Percept. **8**(14), ART14 (2010)
19. Robinson, T., Hochberg, M., Renals, S.: The Use of Recurrent Networks in Continuous Speech Recognition, Chapter 7, pp. 233–258. Kluwer, Boston (1996)
20. Stouten, F., Duchateau, J., Martens, J.-P., Wambacq, P.: Coping with disfluencies in spontaneous speech recognition: acoustic detection and linguistic context manipulation. Speech Commun. **48**, 1590–1606 (2006)
21. van Leeuwen, D.A.: Evaluation plan for the North- and south-dutch Benchmark Evaluation of Speech recognition Technology (N-Best 2008). http://speech.tm.tno.nl/n-best/eval/evalplan.pdf (2008)
22. van Leeuwen, D.A., Kessens, J., Sanders, E., van den Heuvel, H.: Results of the N-Best 2008 Dutch speech recognition evaluation. In: Proceedings of the Interspeech, Brighton, Sept. 2009, pp. 2571–2574. ISCA (2009)
23. van Leeuwen, D.A., Martin, A.F., Przybocki, M.A., Bouten, J.S.: NIST and TNO-NFI evaluations of automatic speaker recognition. Comput. Speech Lang. **20**, 128–158 (2006)
24. Young, S.J., Adda-Dekker, M., Aubert, X., Dugast, C., Gauvain, J.-L., Kershaw, D.J., Lamel, L., van Leeuwen, D.A., Pye, D., Robinson, A.J., Steeneken, H.J.M., Woodland, P.C.: Mutilingual large vocabulary speech recognition: the European SQALE project. Comput. Speech Lang. **11**, 73–89 (1997)

Chapter 16
Missing Data Solutions for Robust Speech Recognition

Yujun Wang, Jort F. Gemmeke, Kris Demuynck, and Hugo Van hamme

16.1 Introduction

One of the major concerns when deploying speech recognition applications is the lack of robustness of the technology. Humans are robust to noise, different acoustic environments, pronunciation variation, ungrammatical sentences, incomplete utterances, filled pauses, stutters, etc. and this engenders the same expectation for automatic systems. In this contribution we discuss an approach called missing data techniques (MDT) [3, 27] to deal with one of these problems: noise robustness. Unlike many previously proposed solutions, MDT can deal with noise exhibiting rapidly changing characteristics, which is often the case in practical deployments. For example, a mobile device used in a city will pick up the noise of cars passing by, of construction sites, from car horns, of people talking or shouting, etc.

In a nutshell, MDT is based on the idea that even in noisy speech, some of the features describing the speech signal remain uncorrupted. The goal is to identify the corrupted (missing) features and to then replace them (impute) with clean speech estimates. In this contribution we describe the research carried out in the MIDAS project, which focussed on two aspects of MDT. First, we discuss an novel imputation method to derive clean speech estimates of the corrupted noise speech features, a method dubbed Sparse Imputation. This method models speech as a linear combination of *exemplars*, segments of speech, rather than modelling speech using a statistical model. Second, we describe how a state-of-the-art large vocabulary automatic speech recognition (ASR) system based on the prevailing hidden Markov model (HMM) can be made noise robust using conventional MDT. Unlike many publications on noise robust ASR, which only report results on

Y. Wang · J.F. Gemmeke · K. Demuynck · H. Van hamme (✉)
ESAT Department, Katholieke Universiteit, Leuven, Belgium
e-mail: yujun.wang@esat.kuleuven.be; jort.gemmeke@esat.kuleuven.be; kris.demuynck@esat.kuleuven.be; hugo.vanhamme@esat.kuleuven.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_16,

artificially corrupted noisy speech, this chapter also describes results on noisy speech recorded in realistic environments.

The rest of the chapter is organised as follows. In Sect. 16.2 we briefly introduce MDT. In Sect. 16.3 we describe the sparse imputation method and the AURORA-2 and Finnish SPEECON [21] databases used for evaluations, and in Sect. 16.4 we describe and discuss the recognition accuracies that were obtained. In Sect. 16.5 we describe the large-vocabulary ASR system, the MDT method employed, and the material from the Flemish SPEECON [21] and SpeechDat-Car [30] databases that were used. In Sect. 16.6 we investigate the performance of the resulting system, both in terms of speech recognition accuracy as well as in terms of speed of program execution. We conclude with a discussion and present our plans for future work in Sect. 16.7.

16.2 Missing Data Techniques

In ASR, the basic representation of speech is a spectro-temporal distribution of acoustic power, a *spectrogram*. The spectrogram typically consist of 20–25 band-pass filters equally spaced on a Mel-frequency scale, and is typically sampled at 8 or 10 ms intervals (a frame). In noise-free conditions, the value of each time-frequency cell in this two-dimensional matrix is determined only by the speech signal. In noisy conditions, the value in each cell represents a combination of speech and background noise power. To mimic human hearing, a logarithmic compression of the power scale is employed.

In the spectrogram of noisy speech, MDT distinguishes time-frequency cells that predominantly contain speech or noise energy by introducing a missing data mask. The elements of that mask are either 1, meaning that the corresponding element of the noisy speech spectrogram is dominated by speech ('reliable') or 0, meaning that it is dominated by noise ('unreliable' c.q. 'missing'). Assuming that only additive noise corrupted the clean speech, the power spectrogram of noisy speech can be approximately described as the sum of the individual power spectrograms of clean speech and noise. As a consequence, in the logarithmic domain, the reliable noisy speech features remain approximately uncorrupted [27] and can be used directly as estimates of the clean speech features. It is the goal of the imputation method to replace ('impute') the unreliable features by clean speech estimates.

After imputation in the Mel-spectral domain, the imputed spectra can be converted to features such as Mel-frequency cepstral coefficients (MFCC). Then, delta and delta-delta derivative features (used in all experiments described in this chapter) can be derived from these. If the clean speech and noise signals or their spectral representations are available so that we know the speech and noise power in each time-frequency cell, a so-called *oracle mask* may be constructed. In realistic situations, however, the location of reliable and unreliable components needs to be estimated. This results in an *estimated mask*. For an overview of mask estimation methods we refer the reader to [2].

16.3 Material and Methods: Sparse Imputation

16.3.1 Sparse Imputation

In this section we give a brief and informal account of the sparse imputation method. For a more formal and in-depth explanation we refer to [12, 15].

In the sparse imputation approach, speech signals are represented as a linear combination of example (clean) speech signals. This linear combination of *exemplars* is sparse, meaning that only a few exemplars should suffice to model the speech signal with sufficient accuracy. The collection of clean speech exemplars used to represent speech signals is called the exemplar dictionary, and is randomly extracted from a training database.

The observed noisy speech signals are processed using overlapping windows, each consisting of a spectrogram spanning 5–30 frames. For this research neighbouring windows were shifted by a single frame. Sparse imputation works in two steps. First, for each observed noisy speech window, a maximally sparse linear combination of exemplars from the dictionary is sought using only the reliable features of the noisy speech and the corresponding features of the exemplars. Then, given this sparse representation of the speech, a clean speech estimate of the unreliable features is made by reconstruction using only those features in the clean speech dictionary that correspond to the locations of the unreliable features of the noisy speech. Applying this procedure for every window position, the clean speech estimates for overlapping windows are combined through averaging.

16.3.2 Databases

The main experiments with sparse imputation have been carried out using the AURORA-2 connected digit database [18] and the Finnish SPEECON large vocabulary database [21]. The use of these databases in [15] and [12] are briefly introduced below. For evaluations of sparse imputation on other tasks and databases, we refer the reader to [9, 10, 23].

In [15], a digit classification task was evaluated using material from the AURORA-2 corpus. The AURORA-2 corpus contains utterances with the digits ‘zero’ through ‘nine’ and ‘oh’, and one to seven digits per utterance. The isolated-digit speech data was created by extracting individual digits using segmentations obtained by a forced alignment of the clean speech utterances with the reference transcription. The clean speech training set of AURORA-2 consists of 27, 748 digits. The test digits were extracted from test set A, which comprises 4 clean and 24 noisy subsets. The noisy subsets are composed of four noise types (subway, car, babble, exhibition hall) artificially mixed at six SNR values, SNR$=$ 20, 15, 10, 5, 0, -5 dB. Every SNR subset consisted of 3,257, 3,308, 3,353 and 3,241 digits per noise type, respectively.

In [12], material from the Finnish SPEECON large vocabulary database was used. The artificially corrupted read speech was constructed by mixing headset-recorded clean speech utterances with a randomly selected sample of the babble noise from the NOISEX-92 database [37] at four SNR values, SNR$=$ 15, 10, 5, 0 dB. The training data consists of 30 h of clean speech recorded with a headset in quiet conditions, spoken by 293 speakers. The test set contains 115 min of speech in 1,093 utterances, spoken by 40 speakers.

16.4 Experiments: Sparse Imputation

In this section we give an overview of the most important results obtained with sparse imputation as reported in [12, 15]. In Sect. 16.4.1 we give a summary of the experimental setup and in Sect. 16.4.2 we discuss the obtained results.

16.4.1 Experimental Setup

16.4.1.1 Digit Classification

For the digit classification task described above, only a single sparse representation was used to represent the entire digit. In other words, only a single window was used, and each digit was time-normalised using linear interpolation to have a fixed length of 35 (8 ms) frames. With each spectrogram consisting of 23 Mel-frequency bands, each digit was thus described by $23 \cdot 35 = 805$ features. The exemplar dictionary consisted of $4,000$ exemplars randomly extracted from the digits in the training database.

Recognition was done using a MATLAB-based ASR engine that can optionally perform missing data imputation using Gaussian-dependent imputation (cf. Sect. 16.5.1) [32]. After applying sparse imputation in the mel-spectral domain, recognition was carried out using PROSPECT features [32]. This technique is described in more detail in Sect. 16.5.1. For further comparison, the cluster-based imputation technique proposed in [26] was used. Two missing data mask methods were used, the oracle mask described in Sect. 16.2 and an estimated mask. In brief, the estimated mask combines a harmonic decomposition and an SNR estimate to label features mostly dominated by harmonic energy and/or with a high SNR as reliable [33].

16.4.1.2 Large Vocabulary Task

For the large vocabulary recognition task using SPEECON, we focus on the results obtained using sparse imputation with spectrograms spanning 20 (8 ms) frames.

With each spectrogram consisting of 21 Mel-frequency bands, each window was thus described by $21 \cdot 20 = 420$ features. The exemplar dictionary consisted of $8,000$ spectrograms randomly extracted from the clean speech in the training database and thus contains anything from whole words, parts of words, word-word transitions to silence and silence-word transitions.

Recognition was done using the large vocabulary continuous speech recognition system developed at the Aalto University School of Science [19]. After imputation in the mel-spectral domain, recognition was carried out using MFCC features. For comparison, the cluster-based imputation technique proposed in [26] was used. The Gaussian-dependent imputation technique used in the other experiments in this chapter was not used, since that method requires recogniser modifications that have not been applied to the Finnish ASR engine used in this experiment. Two missing data mask methods were used, the oracle mask described above and an estimated mask. Unlike the mask estimation method described above, the estimated mask does not employ harmonicty and is constructed using only local SNR estimates obtained from comparing the noisy speech to a static noise estimate calculated during speech pauses [28]. All parameters were optimised using the development data. The speech recognition performance is measured in letter error rates (LER) because the words in Finnish are often very long and consist of several morphemes.

16.4.2 Results

In the experiments described here, the aim is to evaluate the effectiveness of sparse imputation compared to other imputation methods. To that end we compare classification accuracy or recognition accuracy as a function of SNR as obtained with various methods.

In Fig. 16.1 we compare the performance obtained with sparse imputation, cluster-based and Gaussian-dependent imputation on the digit classification task. For estimated masks, we can observe that sparse imputation performed comparably or somewhat worse than Gaussian-dependent imputation but better than cluster-based imputation. For oracle masks, sparse imputation outperforms both Gaussian-dependent and cluster-based imputation by a large margin at SNRs <15 dB.

In Fig. 16.2 we compare the performance obtained with sparse imputation, cluster-based and the baseline recogniser on the SPEECON large vocabulary recognition task. We can observe that sparse imputation performs much better than cluster-based imputation if an oracle mask was used. When using an estimated mask, sparse imputation performed better than cluster-based imputation at lower SNRs, and comparably at higher SNRs. These findings were confirmed in experiments on noisy speech recorded in real-world car and public environments [12].

From these results it is already apparent that advances in mask estimation quality are necessary for further advances in noise robustness, especially for sparse imputation. We will revisit this issue in Sect. 16.7.

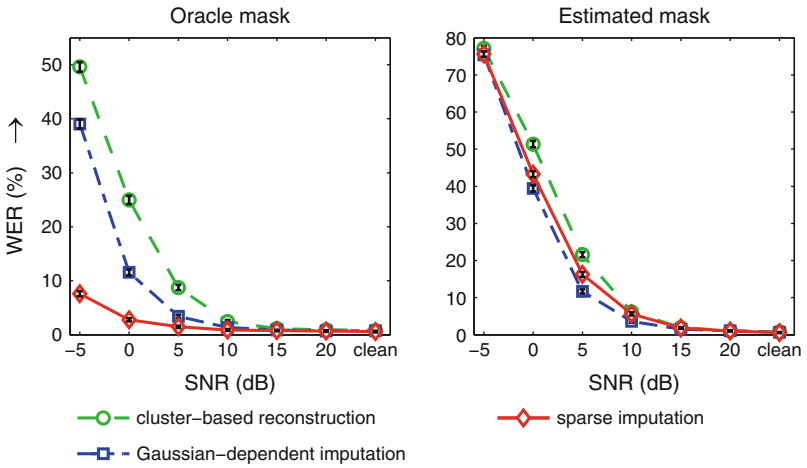


Fig. 16.1 Word error rates (WER) obtained on AURORA-2 isolated digits database with cluster-based, Gaussian-dependent, and sparse imputation. The *left panel* shows the results obtained using oracle masks and the *right panel* shows the results obtained using estimated masks. The horizontal axis describes the SNR at which the clean speech is mixed with the background noise and the vertical axis describes the WER averaged over the four noise types: subway, car, babble, and exhibition hall noise. The vertical bars around data points indicate the 95 % confidence intervals, assuming a binomial distribution

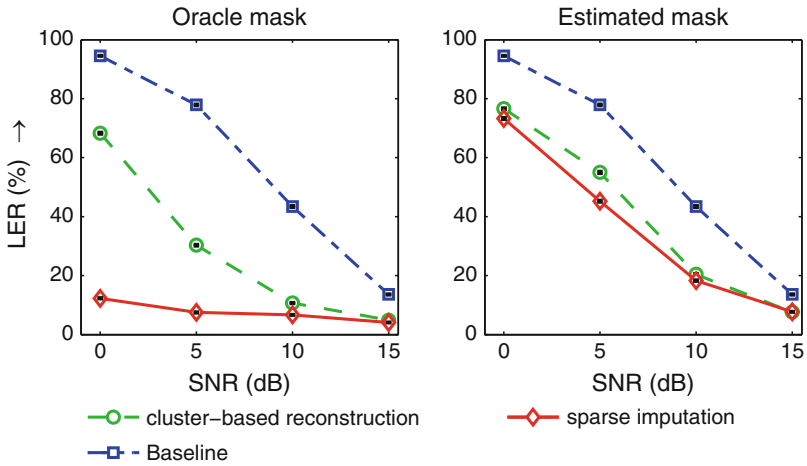


Fig. 16.2 Letter error rates (LER) obtained on the Finnish SPEECON database with cluster-based imputation and sparse imputation. The *left panel* shows the results obtained using oracle masks and the *right panel* shows the results obtained using estimated masks. The horizontal axis describes the SNR at which the clean speech is artificially mixed with babble noise. The vertical bars around data points indicate the 95 % confidence intervals, assuming a binomial distribution

16.5 Material and Methods: Gaussian-Dependent Imputation

16.5.1 Gaussian-Dependent Imputation

Originally, MDT was formulated in the log spectral domain [3]. Here, speech is represented by the log-energy outputs of a filter bank and modelled by a Gaussian Mixture Model (GMM) with diagonal covariance. In the *imputation* approach to MDT, the GMM is then used to reconstruct clean speech estimates for the unreliable features. When doing *bounded imputation*, the unreliable features are not discarded but used as an upper bound on the log-power of the clean speech estimate [4].

Later, it was found the method could be improved by using state-dependent [22] or even Gaussian-dependent [31] clean speech estimates. In these approaches, the unreliable features are imputed during decoding and effectively depend on the hypothesised state identity. However, filter bank outputs are highly correlated and poorly modelled with a GMM with a diagonal covariance. This is the reason why conventional (non-MDT) speech recognisers employ cepstral features, obtained by applying a de-correlating Discrete Cosine Transformation (DCT) on the spectral features.

In [31] it was proposed to do cepstral-domain Gaussian-dependent (bounded) imputation by solving a non-negative least squares (NNLSQ) problem. The method proposed in [32] refines that approach by replacing the DCT used in the generation of cepstra by another data-independent linear transformation that results in computational gains while solving the NNLSQ problem. The resulting PROSPECT features are, just like cepstral coefficients, largely uncorrelated and therefore allow to retain the high accuracy at high SNRs as well as the good performance at lower SNRs obtained with Gaussian-dependent imputation.

16.5.1.1 Multi-candidate MDT

In this chapter we use a faster approach that does not solve the imputation problem for every backend Gaussian (BG), the Gaussians of the HMM acoustic model, but only for a small set of Gaussians using a technique called Multi-Candidate(MC) MDT [38]. In MC MDT, a reduced set of Cluster Gaussians (CG) are established on top of the BGs, with the number of CGs one to two orders of magnitude smaller than the number of BGs. Instead of solving the imputation problem for each BG, candidate solutions are selected from the CGs through MDT imputation. The candidate that maximises the likelihood of the BG is retained as the BG-dependent prediction of the clean speech. In other words, the MDT imputation problem is solved approximately for the BG by constraining possible solutions to a set proposed by the CGs. Computational gains in CG imputation are again obtained by a PROSPECT formulation. The imputed clean filter-bank energies are then

transformed to the preferred feature representation of the BGs. This means that the backend acoustic model of a non-MDT system can be used, which constitutes a great advantage when building MDT systems.

However, since there may be hundreds of CGs, it is not feasible to evaluate each BG on each candidate solutions. Therefore, for every BG, we construct a short-list of CGs that were most successful in producing a winning candidate on a forced alignment of clean training data. The length of this short-list controls the trade-off between computational effort and obtained robustness. Experiments have shown that retaining only a handful of CGs does not lead to loss of accuracy.

During recognition, Gaussian selection is combined with MC MDT. Gaussian selection is motivated by the observation that only a small (frame dependent) portion of Gaussians dominate the likelihoods of the HMM states, and are therefore worth evaluating. The likelihood of a CG evaluated at its imputed value is used to select only the CGs that describe the frame of data sufficiently well. The unlikely CGs are not allowed to propose candidates for evaluation by the BG, which leads to the desired result that unlikely BGs are not evaluated. The proposed Gaussian selection method differs from traditional Gaussian selection methods [1, 6] in that it uses MDT to select relevant clusters. This is advantageous since data that is not close to the Gaussian means because it is severely corrupted by noise can still activate the appropriate Gaussian that models the underlying clean speech.

16.5.1.2 Mask Estimation

To estimate the missing data mask, we use a Vector Quantisation (VQ) strategy that is closely related the method employed in [36]. The key idea is to estimate masks by making only weak assumptions about the noise, while relying on a strong model for the speech, captured in a codebook. The harmonicity found in voiced speech is exploited through a harmonic decomposition as proposed in [33], which decomposes the signal in two parts: the periodic signal part consists of the harmonics at pitch multiples and the remaining spectral energy is considered the aperiodic part.

We first construct a codebook of clean speech by clustering stacked features containing framed spectral representations of the periodic and aperiodic decomposition of the clean speech. Since the codebook only represents a model for the human voice, decoding of non-speech (or noise) frames will lead to incorrect codebook matching and misclassification of mask elements. Therefore, a second, much smaller codebook, is used for non-speech frames. A *Voice Activity Detector* (VAD), segments speech from non-speech frames in order to select the appropriate VQ codebook. To compensate for linear channel distortions, the VQ-system self-adjusts the codebook to the channel during recognition.

The input for the VQ-system consists of three components. The first two components, the spectral representation of the periodic and aperiodic decomposition of the noisy speech, match the content of the codebook entries. The third input component is an estimate of the mean of the noise spectrum. The VQ-system

compares the first two components with the content of each codebook entry, given that noise must be added to the noise-free codebook entries. The instantaneous noise is assumed to be drawn from a distribution with the given noise mean (the third input) and is assumed to be smooth, meaning that it has no periodic structure so that the instantaneous noise periodic and aperiodic parts are close to identical. The smoothness assumption is reasonable for many noise types including speech babble, but it may be violated for a single interfering speaker or for some types of music.

The two noise assumptions allow a closed form distance metric for comparing the noise free VQ-entries with the noisy input and, as a side effect, also returns the estimated instantaneous noise [36]. A speech estimate is obtained by summing the periodic and aperiodic both parts of the codebook entry. Once the best matching codebook entry is found, the spectrographic VQ-based mask is estimated by thresholding the ratio of speech and noise estimates.

The noise tracker (third input to the VQ-system) combines two techniques. First, a short-term spectral estimate of the aperiodic noise is obtained from minimum statistics [24] on the aperiodic component of the noisy signal over a sub-second window. This system is well suited for rapid changing noise types with no periodic structure. A disadvantage of this approach is that the tracker also triggers on long fricatives and fails on periodic noise types. Whereas the experiments in previous publications used only this method, in this work we added a noise tracker developed for another noise robust technique present in SPRAAK called *noise normalisation* [7]. This second noise tracker looks over a longer 1.5 s window and uses ordered statistics instead of minimum statistics to obtain more robust and accurate noise estimates. By combining the two noise trackers, good behaviour on both stationary, non-stationary and periodic noise types is obtained.

16.5.2 Real-World Data: The SPEECON and SpeechDat-Car Databases

In the research reported in this part of the chapter we use material from the Flemish SPEECON [21] and the SpeechDat-Car [30] databases. These databases contain speech recorded in realistic environments with multiple microphones. In total, there are four recording environments: office, public hall, entertainment room and car. All speech material was simultaneously recorded with four different microphones (channels) at increasing distances, resulting in utterances corrupted by varying levels of noise. We used the method described in [16] to obtain SNR estimates of all utterances.

The multi-condition training data consists of 231,849 utterances spoken in 205 h of speech. The speech is taken from the office, public hall and car noise environments, with most of the data (168 h) coming from the office environment. It contains a clean data portion of 61,940 utterances from channel #1 (closetalk microphone) data with an estimated SNR range of 15–50 dB. Additionally, the

multi-condition set contains all utterances from channels #2, #3 and #4 which have an estimated SNR of 10 dB and higher, containing 54,381, 53,248 and 31,975 utterances, respectively.

For the test set, we use material pertaining to a connected digit recognition task. The utterances contain the ten digits 'zero' through 'nine', with between one and ten digits per utterance. The 6,218 utterances (containing 25,737 digits) of the test set are divided in 6 SNR subsets in the 0–30 dB range with a 5 dB bin width. The SNR bins do not contain equal numbers of utterances from the four channels: Generally speaking, the highest SNR bins mostly contain utterances from channel #1, while the lowest SNR bins mostly contains channel #4 speech.

16.6 Experiments: Gaussian-Dependent Imputation

16.6.1 Experimental setup

16.6.1.1 Speech Recogniser and Acoustic Models

The implementation of MDT does not require a complete overhaul of the software architecture of a speech recogniser. We extended the code of the SPRAAK-recogniser (described in Chap. 6, p. 95) to include a missing data mask estimator (cf. Sect. 16.5.1.2) and to evaluate the acoustic model according to the principles described in Sect. 16.5.1. Below, we will successively describe the configuration in which SPRAAK was used, how its acoustic models were created, how the baseline so-called PROSPECT models were created to benchmark the proposed speed-ups and how the data structures for the multi-candidate MDT were obtained.

The acoustic feature vectors consisted of MEL-frequency log power spectra: 22 frequency bands with centre frequencies starting at 200 Hz. The spectra were created by framing the 16 kHz signal with a Hamming window with a window size 25 ms and a frame shift of 10 ms. The decoder also uses the first and second time derivative of these features, resulting in a 66-dimensional feature vector. This vector is transformed linearly by using Mutual Information Discriminant Analysis (MIDA) linear transformation [5]. During training, mean normalisation is applied to the features. During decoding, the features are normalised by a more sophisticated technique which is compatible with MDT and which works by updating an initial channel estimate through maximisation of the log-likelihood of the best-scoring state sequence of a recognised utterance [35].

The training of the multi-condition context-dependent acoustic models on a set of 46 phones plus four filler models and a silence model follows the standard training scripts of SPRAAK and leads to 4,476 states tied through a phonetic decision tree and uses a pool of 32,747 Gaussians.

16.6.1.2 Imputation

A set of 700 cluster Gaussians required for the MC-MDT acoustic model was obtained by pruning back the phonetic tree to 700 leaves, each modelled with a single PROSPECT Gaussian trained on the respective training sets. The cluster Gaussians and backend Gaussians are associated in a table which retains only the most frequent co-occurrence of the most likely cluster Gaussian and the most likely backend Gaussian in Viterbi alignment of the training data. The SPRAAK toolkit was extended with adequate tools to perform these operations. The association table was then pruned to allow maximally five cluster Gaussians per back-end Gaussian. The average number of cluster Gaussians per back-end Gaussian is 3.6.

The VQ-codebook used in mask estimation was trained on features extracted from the close-talk channel SPEECON training database. The number of codebook entries was 500 for speech and 20 for silence. Recognition tests on the complete test set using a large interval of threshold values revealed that the threshold setting was not very sensitive. The (optimal) results presented in this work were obtained with 8 dB. Missing data masks for the derivative features were created by taking the first and second derivative of the missing data mask [34].

16.6.1.3 VOCON

The VOCON 3200 ASR engine is a small-footprint engine, using MFCC based features and HMM models. It contains techniques to cope with stationary or slowly varying background noise. Its training data includes in-car recorded samples, i.e., it uses the multi-condition training approach in tandem with noise reduction techniques. The VOCON recogniser uses whole-word models to model digits, whereas the MDT system uses triphones.

16.6.2 Results

In Fig. 16.3 we compare the performance obtained with the SPRAAK baseline system (the SPRAAK system described in Sect. 16.6.1.1, without employing MDT), the SPRAAK MDT system and the VOCON recogniser. We can observe that the use of MDT in the SPRAAK recogniser reduces the WER substantially in all noise environments and at all SNRs. The only exception is the 0–5 dB SNR bin in the office noise environment, but here the difference with the SPRAAK baseline is not significant. When comparing the SPRAAK recognisers with the VOCON recogniser, we observe that the VOCON recogniser typically performs better at the lowest SNRs, but at the cost of a higher WER at higher SNRs.

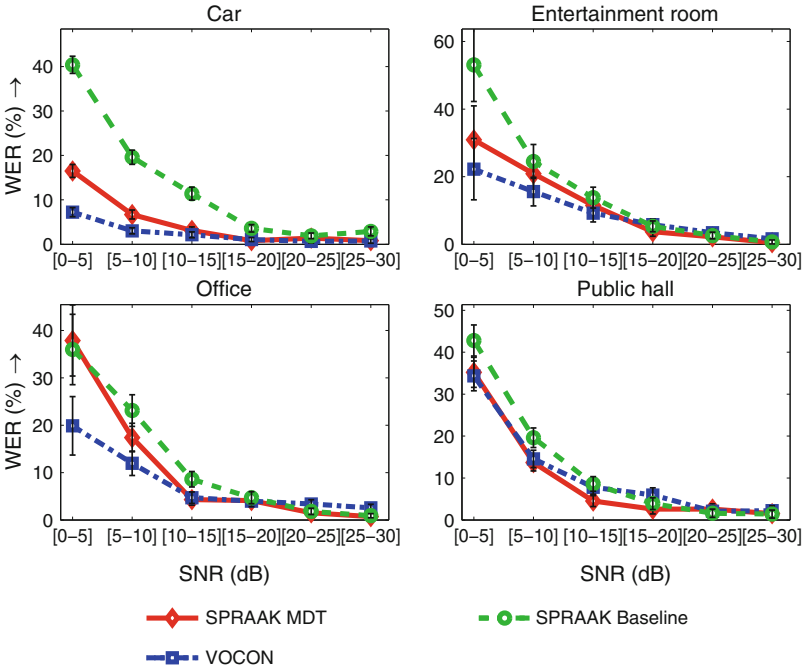


Fig. 16.3 Word error rates (WER) obtained on the Flemish SPEECON database on a connected digit recognition task, comparing the SPRAAK recogniser using imputation (MDT), the SPRAAK baseline and the VOCON recogniser. Four noise environments are shown, viz. car, entertainment room, office, and public hall. The horizontal axis describes the estimated SNR of the noisy speech. The vertical bars around data points indicate the 95 % confidence intervals, assuming a binomial distribution

Table 16.1 Timing experiments using speech from the 25–30 dB SNR bin

	CPU time (ms/frame)						
	Mask	BG	CG	Beam search	Total	BGs calculated (%)	WER (%)
SPRAAK baseline	0.0	0.9	0.2	1.8	2.9	5	1.47
SPRAAK MDT	1.1	4.1	4.2	1.8	11.2	14	0.88

In Tables 16.1 and 16.2 we show the results of a timing experiment on 'clean' speech (25–30 dB SNR) and noisy speech (10–15 dB SNR), respectively. The timings are obtained by recognition of 10 randomly selected sentences per noise environment (40 in total), which together contain 22,761 frames for the clean speech and 16,147 frames for the noisy speech. We can observe that the use of MDT in SPRAAK is approximately four times slower than the SPRAAK baseline in clean

Table 16.2 Timing experiments using speech from the 10–15 dB SNR bin

	CPU time (ms/frame)						
	Mask	BG	CG	Beam search	Total	BGs calculated (%)	WER (%)
SPRAAK baseline	0.0	2.7	0.2	1.9	4.8	13	10.0
SPRAAK MDT	1.1	7.2	4.7	1.8	21.1	27	5.85

conditions, but only two times slower in noisy conditions. As can be seen from the average WERs in the two tables, the use of MDT approximately halves the WER, even in the cleaner conditions.

16.7 Discussion and Conclusions

From the results obtained with sparse imputation, one can draw two conclusions. On the one hand, the sparse imputation method achieved impressive reductions in WER and LER when used in combination with an oracle mask. On the other hand, although sparse imputation performs better than cluster-based imputation when using estimated masks, it does not perform better than Gaussian-dependent imputation. This means that for sparse imputation to reach its full potential, advances in mask estimation techniques are necessary. Unfortunately, despite a decade of research on missing data techniques the gap between estimated masks and oracle masks remains [8].

From the results obtained with the SPRAAK recogniser employing MDT, we observed a substantial improvement in noise robustness. Although at the cost of two to four times lower execution speed, the WER halved even in the cleaner conditions. Moreover, it was reported in [38] that the proposed MDT technique is not significantly slower than the baseline SPRAAK recogniser when applied on a large vocabulary task. In comparison to the VOCON recogniser, the SPRAAK recogniser typically performs better at moderate-to-high SNRs. The noise robustness of the VOCON recogniser at low SNRs can probably be attributed to its use of whole-word models.

With respect to sparse imputation, various improvements have been proposed recently, such as the use of *probabilistic masks* [11], the use of observation uncertainties to make the recogniser aware of errors in estimating clean speech features [14], and the use of additional constraints when finding a sparse representation [29]. Finally, in the course of the MIDAS project a novel speech recognition method was proposed which explicitly models noisy speech as a sparse linear combination of speech and noise exemplars, thus bypassing the need for a missing data mask. Although only evaluated on small vocabulary tasks, the results are promising [13, 17, 20], e.g., achieving a 37.6 % WER at SNR $= -5$ dB on AURORA-2.

Future work concerning the noise robust SPRAAK recogniser will focus on improving mask estimation quality in two ways. First, while it has been shown MDT can be used to combat reverberation [16, 25], to date no method has been

presented that enables the estimation of reverberation-dominated features in noisy environments. Second, future work will address the poor performance of current mask estimation methods on speech corrupted by background music, a prevailing problem in searching audion archives.

References

1. Bocchieri, E.: Vector quantization for efficient computation of continuous density likelihoods. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, vol. 2, Minneapolis, Minnesota, USA, pp. 692–695 (1993)
2. Cerisara, C., Demange, S., Haton, J.P.: On noise masking for automatic missing data speech recognition: A survey and discussion. Comput. Speech Lang. **21**(3), 443–457 (2007)
3. Cooke, M., Green, P., Crawford, M.: Handling missing data in speech recognition. In: Proceedings of the International Conference on Spoken Language Processing, Yokohama, Japan, pp. 1555–1558 (1994)
4. Cooke, M., Green, P., Josifovski, L., Vizinho, A.: Robust automatic speech recognition with missing and unreliable acoustic data. Speech Commun. **34**(3), 267–285 (2001)
5. Demuynck, K., Duchateau, J., Compernolle, D.V.: Optimal feature sub-space selection based on discriminant analysis. In: Proceedings of the European Conference on Speech Communication and Technology, vol. 3, Budapest, Hungary, pp. 1311–1314 (1999)
6. Demuynck, K., Duchateau, J., Van Compernolle, D.: Reduced semi-continuous models for large vocabulary continuous speech recognition in Dutch. In: Proc. the International Conference on Spoken Language Processing, vol. IV, Philadelphia, USA, pp. 2289–2292 (1996)
7. Demuynck, K., Zhang, X., Van Compernolle, D., Van hamme, H.: Feature versus model based noise robustness. In: Proc. INTERSPEECH, Makuhari, Japan, pp. 721–724 (2010)
8. Gemmeke, J.F.: Noise robust ASR: missing data techniques and beyond. Ph.D. Thesis, Radboud Universiteit Nijmegen, The Netherlands (2011)
9. Gemmeke, J.F., Cranen, B.: Noise reduction through compressed sensing. In: Proceedings of the INTERSPEECH, Brisbane, Australia, pp. 1785–1788 (2008)
10. Gemmeke, J.F., Cranen, B.: Missing data imputation using compressive sensing techniques for connected digit recognition. In: Proceedings of the International Conference on Digital Signal Processing, Santorini, Greece, pp. 1–8 (2009)
11. Gemmeke, J.F., Cranen, B.: Sparse imputation for noise robust speech recognition using soft masks. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, pp. 4645–4648 (2009)
12. Gemmeke, J.F., Cranen, B., Remes, U.: Sparse imputation for large vocabulary noise robust ASR. Comput. Speech Lang. **25**(2), 462–479 (2011)
13. Gemmeke, J.F., Hurmalainen, A., Virtanen, T., Sun, Y.: Toward a practical implementation of exemplar-based noise robust ASR. In: Proceedings of the EUSIPCO, Barcelona, Spain, pp. 1490–1494 (2011)
14. Gemmeke, J.F., Remes, U., Palomäki, K.J.: Observation uncertainty measures for sparse imputation. In: Proceedings of the Interspeech, Makuhari, Japan, pp. 2262–2265 (2010)
15. Gemmeke, J.F., Van hamme, H., Cranen, B., Boves, L.: Compressive sensing for missing data imputation in noise robust speech recognition. IEEE J Sel. Top. Signal Process. **4**(2), 272–287 (2010)

16. Gemmeke, J.F., Van Segbroeck, M., Wang, Y., Cranen, B., Van hamme, H.: Automatic speech recognition using missing data techniques: handling of real-world data. In: Kolossa, D., Haeb-Umbach R. (eds.) Robust Speech Recognition of Uncertain or Missing Data, pp. 157–185. Springer Verlag, Berlin-Heidelberg (Germany) (2011)
17. Gemmeke, J.F., Virtanen, T., Hurmalainen, A.: Exemplar-based sparse representations for noise robust automatic speech recognition. IEEE Trans. Audio Speech Lang. process. **19**(7), 2067–2080 (2011)
18. Hirsch, H., Pearce, D.: The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In: Proceedings of the ISCA Tutorial and Research Workshop ASR2000, Paris, France, pp. 181–188 (2000)
19. Hirsimäki, T., Creutz, M., Siivola, V., Kurimo, M., Virpioja, S., Pylkkönen, J.: Unlimited vocabulary speech recognition with morph language models applied to Finnish. Comput. Speech Lang. **20**(4), 515–541 (2006)
20. Hurmalainen, A., Mahkonen, K., Gemmeke, J.F., Virtanen, T.: Exemplar-based recognition of speech in highly variable noise. In: International Workshop on Machine Listening in Multisource Environments, Florence, Italy (2011)
21. Iskra, D., Grosskopf, B., Marasek, K., van den Heuvel, H., Diehl, F., Kiessling, A.: Speecon – speech databases for consumer devices: Database specification and validation. In: Proceedings of the of LREC, Las Palmas, Spain, pp. 329–333 (2002)
22. Josifovski, L., Cooke, M., Green, P., Vizinho, A.: State based imputation of missing data for robust speech recognition and speech enhancement. In: Proceedings of the EUROSPEECH, Budapest, Hungary, pp. 2837–2840 (1999)
23. Kallasjoki, H., Keronen, S., Brown, G.J., Gemmeke, J.F., Remes, U., Palomäki, K.J.: Mask estimation and sparse imputation for missing data speech recognition in multisource reverberant environments. In: International Workshop on Machine Listening in Multisource Environments, Florence, Italy (2011)
24. Martin, R.: Noise power spectral density estimation based on optimal smoothing and minimum statistics. IEEE Trans. Speech Audio Process. **9**, 504–512 (2001)
25. Palomäki, K.J., Brown, G.J., Barker, J.: Techniques for handling convolutional distortion with "missing data" automatic speech recognition. Speech Commun. **43**, 123–142 (2004)
26. Raj, B., Seltzer, M.L., Stern, R.M.: Reconstruction of missing features for robust speech recognition. Speech Commun. **43**(4), 275–296 (2004)
27. Raj, B., Stern, R.M.: Missing-feature approaches in speech recognition. IEEE Signal Process. Mag. **22**(5), 101–116 (2005)
28. Remes, U., Palomäki, K.J., Kurimo, M.: Missing feature reconstruction and acoustic model adaptation combined for large vocabulary continuous speech recognition. In: Proceedings of the EUSIPCO, Lausanne, Switzerland (2008)
29. Tan, Q.F., Georgiou, P.G., Narayanan, S.S.: Enhanced sparse imputation techniques for a robust speech recognition front-end. IEEE Trans Audio Speech Lang. Process. **19**(8), 2418–2429 (2011)
30. van den Heuvel, H., Boudy, J., Comeyne, R., Communications, M.N.: The speechdat-car multilingual speech databases for in-car applications. In: Proceedings of the European Conference on Speech Communication and Technology, Budapest, Hungary, pp. 2279–2282 (1999)
31. Van hamme, H.: Robust speech recognition using missing feature theory in the cepstral or LDA domain. In: Proceedings of the EUROSPEECH, Geneva, Switzerland, pp. 3089–3092 (2003)
32. Van hamme, H.: PROSPECT features and their application to missing data techniques for robust speech recognition. In: Proceedings of the INTERSPEECH, Jeju Island, Korea, pp. 101–104 (2004)
33. Van hamme, H.: Robust speech recognition using cepstral domain missing data techniques and noisy masks. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Montreal, Quebec, Canada, pp. 213–216 (2004)

34. Van hamme, H.: Handling time-derivative features in a missing data framework for robust automatic speech recognition. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Toulouse, France (2006)
35. Van Segbroeck, M., Van hamme, H.: Handling convolutional noise in missing data automatic speech recognition. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, pp. 2562–2565 (2006)
36. Van Segbroeck, M., Van hamme, H.: Vector-Quantization based mask estimation for missing data automatic speech recognition. In: Proceedings of the INTERSPEECH, Antwerp, Belgium, pp. 910–913. (2007)
37. Varga, A., Steeneken, H.: Assessment for automatic speech recognition: II. NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems. Speech Commun. **12**(3), 247–51 (1993)
38. Wang, Y., Van hamme, H.: Multi-candidate missing data imputation for robust speech recognition. EURASIP Journal on Audio, Speech, and Music Processing, No. 17, doi:10.1186/1687-4722-2012-17, May 2012

Chapter 17
Parse and Corpus-Based Machine Translation

Vincent Vandeghinste, Scott Martens, Gideon Kotzé, Jörg Tiedemann, Joachim Van den Bogaert, Koen De Smet, Frank Van Eynde, and Gertjan van Noord

17.1 Introduction

The current state-of-the-art in machine translation consists of *phrase-based statistical machine translation (PB-SMT)* [23], an approach which has been used since the late 1990s, evolving from word-based SMT proposed by IBM [5]. These *string-based* techniques (which use no linguistic knowledge) seem to have reached their ceiling in terms of translation quality, while there are still a number of limitations to the model. It lacks a mechanism to deal with long-distance dependencies, it has no means to generalise over non-overt linguistic information [37] and it has limited word reordering capabilities. Furthermore, in some cases the output quality may lack appropriate fluency and grammaticality to be acceptable for actual MT users. Sometimes essential words are missing from the translation.

To overcome these limitations efforts have been made to introduce *syntactic knowledge* into the statistical paradigm, usually in the form of syntax trees, either

V. Vandeghinste (✉) · J. Van den Bogaert · F. Van Eynde
Centrum voor Computerlinguïstiek (CCL), Leuven University, Leuven, Belgium
e-mail: vincent@ccl.kuleuven.be; joachim@ccl.kuleuven.be; frank.vaneynde@ccl.kuleuven.be

S. Martens
University of Tübingen (previously at CCL), Tübingen, Germany
e-mail: scott.martens@uni-tuebingen.de

G. Kotzé · G. van Noord
Groningen University, Groningen, The Netherlands
e-mail: g.j.kotze@rug.nl; g.j.m.van.noord@rug.nl

J. Tiedemann
University of Uppsala (previously at Groningen University), Uppsala, Sweden
e-mail: jorg.tiedemann@lingfil.uu.se

K. De Smet
Oneliner bvba, Sint-Niklaas, Belgium
e-mail: koen@oneliner.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_17,

only for the source (tree-to-string) or the target language (string-to-tree), or for both (tree-to-tree).

Galley et al. [12] describes an MT engine in which *tree-to-string* rules have been derived from a parallel corpus, driven by the problems of SMT systems raised by [11]. Marcu et al. and Wang et al. [30, 52] describe *string-to-tree* systems to allow for better reordering than phrase-based SMT and to improve grammaticality. Hassan et al. [18] implements another string-to-tree system by means of including supertags [2] to the target side of the phrase-based SMT baseline.

Most of the *tree-to-tree* approaches use one or another form of *synchronous context-free grammars* (SCFGs) a.k.a. syntax directed translations [1] or syntax directed *transduction* grammars [28]. This is true for the tree-based models of the Moses toolkit,[1] and the machine translation techniques described in, amongst others [7,27,36,53–55]. A more complex type of translation grammars is *synchronous tree substitution grammar (STSG)* [10, 38] which provides a way, as [8] points out, to perform certain operations which are not possible with SCFGs without flattening the trees, such as raising and lowering nodes. Examples of STSG approaches are the *Data-Oriented Translation (DOT)* model from [20,35] which uses data-oriented parsing [3] and the approaches described in [14–16] and [37], using STSG rules consisting of dependency subtrees, and a top-down transduction model using beam search.

The *Parse and Corpus based MT (PaCo-MT)* engine described in this chapter[2] is another tree-to-tree system that uses an STSG, differing from related work with STSGs in that the PaCo-MT engine combines dependency information with constituency information and that the translation model abstracts over word and phrase order in the synchronous grammar rules: the daughters of any node are in a canonical order representing all permutations. The final word order is generated by the tree-based target language modeling component.

Figure 17.1 presents the architecture of the PaCo-MT system. A source language (SL) sentence gets syntactically analysed by a pre-existing parser which leads to a source language parse tree, making abstraction of the surface order. This is described in Sect. 17.2. The unordered parse tree is translated into a forest of unordered trees (a.k.a. bag of bags) by applying *tree transduction* with the *transfer grammar* which is an STSG derived from a parallel treebank. Section 17.3 presents how the transduction grammar was built and Sect. 17.4 how this grammar is used in the translation process. The forest is decoded by the *target language generator*, described in Sect. 17.5 which generates an n-best list of translation alternatives by using a tree-based target language model. The system is evaluated on Dutch *to* English in Sect. 17.6 and conclusions are drawn in Sect. 17.7. As all modules of our system are language independent results for Dutch → French, English → Dutch, and French → Dutch can be expected soon.

[1] http://www.statmt.org/moses/

[2] Previous versions were described in [48] and [49].

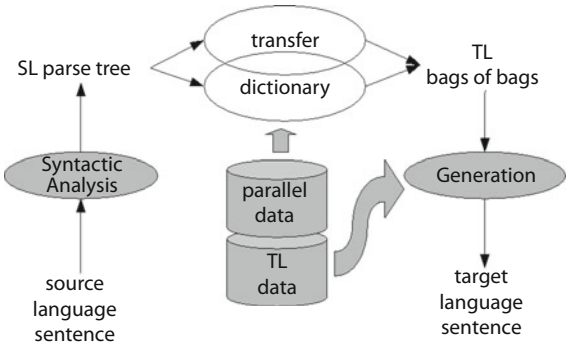


Fig. 17.1 The architecture of the PaCo-MT system

17.2 Syntactic Analysis

Dutch input sentences are parsed using Alpino [32], a stochastic rule-based dependency parser, resulting in structures as in Fig. 17.2.[3]

In order to induce the translation grammar, as explained in Sect. 17.3, parse trees for the English sentences in the parallel corpora are also required. These sentences are parsed using the Stanford phrase structure parser [21] with dependency information [31]. The bracketed phrase structure and the typed dependency information are integrated into an XML format consistent with the Alpino XML format. All tokens are lemmatised using TreeTagger [39].

Abstraction is made of the surface order of the terminals in every parse tree used in the PaCo-MT system. An *unordered tree* is defined[4] by the tuple $\langle V, V^i, E, L\rangle$ where V is the set of nodes, V^i is the set of internal nodes, and $V^f = V - V^i$ is the set of frontier nodes, i.e. nodes without daughters. $E \subset V^i \times V$ is the set of directed edges and L is the set of labels on nodes or edges. $V^l \subseteq V^f$ is the set of lexical frontier nodes, containing actual words as labels, and $V^n = V^f - V^l$ is the set of non-lexical frontier nodes, which is empty in a full parse tree, but not necessarily in a subtree. There is exactly one root node $r \in V^i$ without incoming edges. Let T be the set of all unordered trees, including subtrees.

A subtree $s_r \in T$ of a tree $t \in T$ has as a root node $r \in V_t^i$ where V_t^i is the set of internal nodes of t. Subtrees are *horizontally complete* [4] if, when a daughter node of a node is included in the subtree, then so are all of its sisters. Figure 17.3 shows an example. Let $H \subset T$ be the set of all horizontally complete subtrees.

Bottom-up subtrees are a subset of the horizontally complete subtrees: they are lexical subtrees: every terminal node of the subtree is a lexical node. Some examples are shown in Fig. 17.4. Let $B \subset H$ be the set of all bottom-up subtrees. $\forall b \in B$:

[3]Limited restructuring is applied to make the resulting parse trees more uniform. For instance, nouns are always placed under an NP. A similar restructuring of syntax trees is shown by [52] to improve translation results.

[4]This definition is inspired by [10].

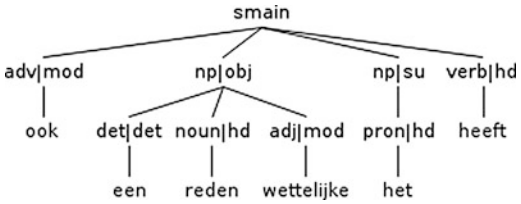


Fig. 17.2 An unordered parse tree for the Dutch sentence *Het heeft ook een wettelijke reden* "It also has a legal reason", or according to Europarl "It is also subject to a legal requirement". Note that edge labels are marked behind the '|'

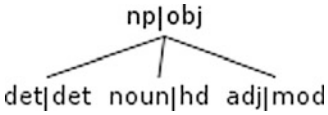


Fig. 17.3 An example of a horizontally complete subtree which is not a bottom-up subtree

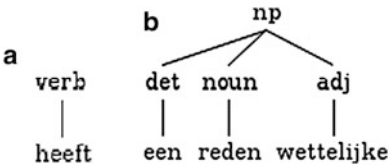


Fig. 17.4 Two examples of bottom-up subtrees

$V_b^n = \emptyset$ and $V_b^l = V_b^f$, where V_b^n is the set of non-lexical frontier nodes of b and V_b^l is the set of lexical frontier nodes of b. V_b^f is the set of all frontier nodes of b.

17.3 The Transduction Grammar

In order to translate a source sentence, a stochastic synchronous tree substitution grammar G is applied to the source sentence parse tree. Every grammar rule $g \in G$ consists of an elementary tree pair, defined by the tuple $\langle d^g, e^g, A^g \rangle$, where $d^g \in T$ is the source side tree (Dutch), $e^g \in T$ is the target side tree (English), and A^g is the alignment between the non-lexical frontier nodes of d^g and e^g. The alignment A^g is defined by a set of tuples $\langle v_d, v_e \rangle$ where $v_d \in V_d^n$ and $v_e \in V_e^n$. V_d^n is the set of non-lexical frontier nodes of d^g, and V_e^n is the set of non-lexical frontier nodes of e^g. Every non-lexical frontier node of the source side is aligned with a non-lexical frontier node of the target side: $\forall v_d \in V_d^n$ is aligned with a node $v_e \in V_e^n$. An example grammar rule is shown in Fig. 17.5.

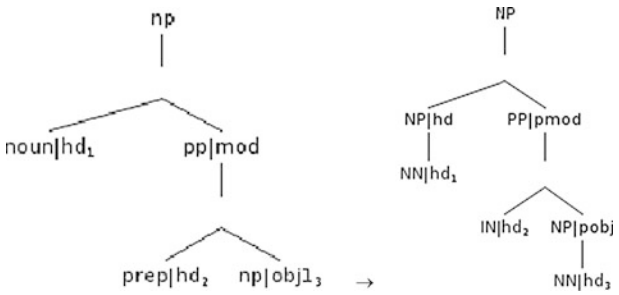


Fig. 17.5 An example of a grammar rule with horizontally complete subtrees on both the source and target side. Indices mark alignments

In order to induce such a grammar a *node aligned* parallel treebank is required. Section 17.3.1 describes how to build such a treebank. Section 17.3.2 describes the actual induction process.

17.3.1 Preprocessing and Alignment of the Parallel Data

The system was trained on the Dutch-English subsets of the Europarl corpus [22], the DGT translation memory,[5] the OPUS corpus[6] [42] and an additional private translation memory (transmem).

The data was syntactically parsed (as described in Sect. 17.2), sentence aligned using Hunalign [50] and word aligned using GIZA++ [33]. The bidirectional GIZA++ word alignments were refined using the *intersect* and *grow-diag* heuristics implemented by Moses [24], resulting in a higher recall for alignments suitable for machine translation.

For training Lingua-Align [43], which is a discriminative tree aligner [44], a set of parallel alignments was manually constructed using the Stockholm TreeAligner [29], for which the already existing word alignments were imported. The recall of the resulting alignments was rather low, even though in constructing the training data a more relaxed version of the well-formedness criteria as proposed by [19] was used.

Various features and parameters have been used in experimentation, training with around 90 % and testing with the rest of the data set. The training data set consists of 140 parallel sentences.

Recent studies in rule-based alignment error correction ([25,26]) show that recall can be significantly increased while retaining a relatively high degree of precision.

[5] http://langtech.jrc.it/DGT-TM.html

[6] http://opus.lingfil.uu.se/

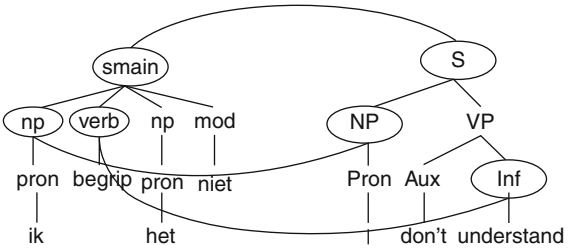


Fig. 17.6 Two sentences with subsentential alignment

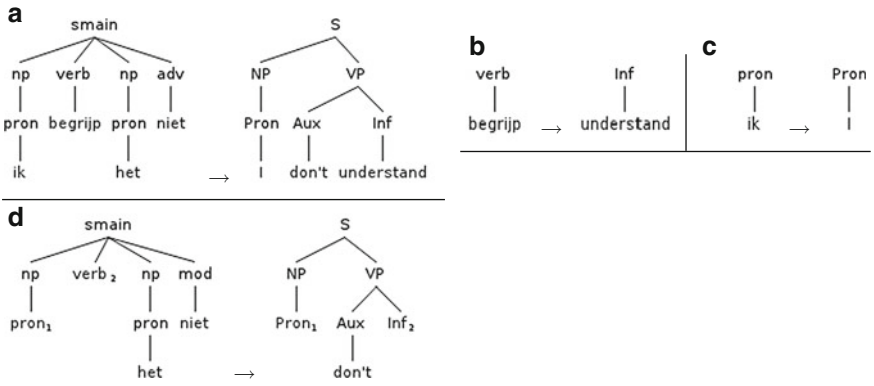


Fig. 17.7 Rules extracted from the alignments in Fig. 17.6

This approach has been extended by applying a bottom-up rule addition component that greedily adds alignments based on already existing word alignments, more relaxed well-formedness criteria, as well as using measures of similarities between the two unlinked subtrees being considered for alignment.

17.3.2 Grammar Rule Induction

Figure 17.6 is an example[7] of two sentences aligned at both the sentence and subsentential level. For each alignment point, either one or two rules are extracted. First, each alignment point is a lexical alignment, creating a rule that maps a source language word or phrase to a target language one (Fig. 17.7a, b).

Secondly, each aligned pair of sentences engenders further rules by partitioning each tree at each alignment point, yielding non-lexical grammar rules. For these

[7]The edge labels have been omitted from these examples, but were used in the actual rule induction.

rules, the alignment information is retained at the leaves so that these trees can be recombined (Fig. 17.7d).

The rule extraction process was restricted to rules with horizontally complete subtrees at the source and target side. Rule extraction with other types of subtrees was considered out of the scope of the current research.

Figure 17.7 shows the four rules extracted from the alignments in Fig. 17.6. Rules are extracted by passing over the entire aligned treebank, identifying each aligned node pair and recursively iterating over its children to generate a substitutable pair of trees whose roots are aligned, and whose leaves are either terminal leaves in the treebank or correspond to aligned vertices. As shown in Fig. 17.7, when a leaf node corresponds to an alignment point, we retain the information to identify which target tree leaf aligns with each such source leaf.

Many such tree substitution rules recur many times in the treebank, and a count is kept of the number of times each pair appears, resulting in a *stochastic* synchronous tree substitution grammar.

17.4 The Transduction Process

The *transduction process* takes an unordered source language parse tree $p \in T$ as input, applies the transduction grammar G and transduces p into an unordered weighted packed forest, which is a compact representation of a set of target trees $Q \subset T$, which represent the translation alternatives. An example of a packed forest is shown in Fig. 17.8.

For every node $v \in V_p^i$, where V_p^i is the set of internal nodes in the input parse tree p, it is checked whether there is a subtree $s_v \in H$ with v as its root node, which matches the source side tree d^g of a grammar rule $g \in G$.

To keep computational complexity limited the subtrees of p that are considered and the subtrees that occur in the source and target side of the grammar G have been restricted to horizontally complete subtrees (including bottom-up subtrees).

When finding a matching grammar rule for which $s_v = d^g$, the corresponding e^g is inserted into the output forest Q. When not finding a matching grammar rule, a horizontally complete subtree is constructed, as explained in Sect. 17.4.2.

The weight that the target side e^g of grammar rule $g \in G$ will get when is calculated according to Eq. 17.1. This weight calculation is similar to the approaches of [14, 37], as it contains largely the same factors. We multiply the weight of the grammar rule $w(g)$ with the relative frequency of the grammar rule over all grammar rules with the same source side $\frac{F(g)}{F(d^g)}$. This is divided by an alignment point penalty $(j+1)^{app}$, favouring the solutions with the least alignment points.

$$W(e^g) = \frac{w(g)}{(j+1)^{app}} \times \frac{F(g)}{F(d^g)} \tag{17.1}$$

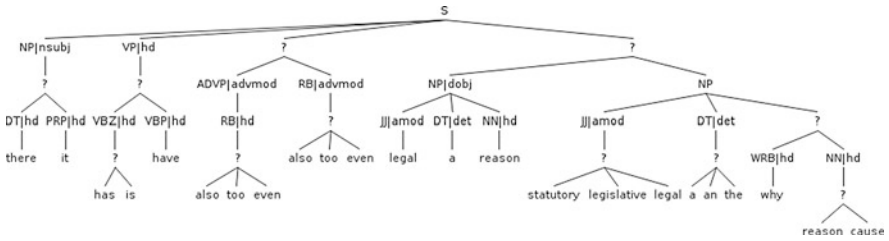


Fig. 17.8 An example of a packed forest as output of the transducer for the Dutch sentence *Het heeft ook een wettelijke reden*. Note that ? marks an alternation

where $w(g) = \sqrt[n]{\prod_{i=1}^{n} w(A_i^g)}$ is the weight of $g \in G$, which is the geometric mean of the weight of each individual occurrence of alignment A, as produced by the discriminative aligner described in Sect. 17.3.1; $j = |V_d^n| = |V_e^n|$ is the number of alignment points, which is the number of non-lexical frontier elements which are aligned in $g \in G$; app is the alignment points power parameter ($app = 0.5$); $F(g)$ is the frequency of occurrence g in the data; $F(d^g)$ is the frequency of occurrence of the source side d of g in the data.

When no translation of a word is found in the transduction grammar, the label $l \in L$ is mapped onto its target language equivalent. Adding a simple bilingual word form dictionary is optional. When a word translation is not found in the transduction grammar, the word is looked up in this dictionary. If the word has multiple translations in the dictionary, each of these translations receives the same weight and is combined with the translated label (usually part-of-speech tags). When the word is not in the dictionary or no dictionary is present, the source word is transfered as is to Q.

17.4.1 Subtree Matching

In a first step, the transducer performs *bottom-up subtree* matching, which is analogous to the use of phrases in phrase-based SMT, but restricted to linguistically meaningful phrases. Bottom-up subtree matching functions like a sub-sentential translation memory: every linguistically meaningful phrase that has been encountered in the data will be considered in the transduction process, obliterating the distinction between a translation memory, a dictionary and a parallel corpus [45].

For every node $v \in V_p$ it is checked whether a subtree s_v with root node v is found for which $s_v \in B$ and for which there is a grammar rule $g \in G$ for which $d = s_v$. These matches include single word translations together with their parts-of-speech.

A second step consists of performing *horizontally complete subtree* matching for those nodes in the source parse tree for which the number of grammar rules $g \in G$ that match is smaller than the beam size b.

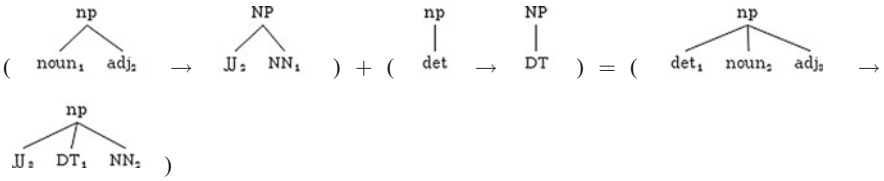


Fig. 17.9 An example of a constructed grammar rule

For every node $v \in V_p^i$ the set $H_v \subset H \setminus B$ is generated, which is the set of all horizontally complete subtrees minus the bottom-up subtrees of p with root node v. It is checked whether a matching subtree $s_v \in H_v$ is found for which there is a grammar rule $g \in G$ for which $d^g = s_v$.

An example of a grammar rule with horizontally complete subtrees on both source and target sides was shown in Fig. 17.5. This rule has three alignment points, as indicated by the indices.

17.4.2 Backing Off to Constructed Horizontally Complete Subtrees

In cases where no grammar rules are found for which the source side matches the horizontally complete subtrees at a certain node in the input parse tree, grammar rules are combined for which, when combined, the source sides form a horizontally complete subtree. An example of such a constructed grammar rule is shown in Fig. 17.9.

$\forall v \in V_p^i$ for which there is no $s_v \in H_v$ matching any grammar rule $g \in G$, let $C_s = \langle c_1, \ldots, c_n \rangle$ be the set of children of root node v in subtree s_v. $\forall c_j \in C_s$ the subtree s_v is split into two partial subtrees y_v and z_v, where $C_y = C_s \setminus \{c_j\}$ is the set of children of subtree y_v and $C_z = \{c_j\}$ is the set of children of subtree z_v.

When a grammar rule $g \in G$ is found for which $d^g = y_v$ and another grammar rule $h \in G$ is found for which $d^h = z_v$, then the respective target sides e_q^g with root node q and e_u^h with root node u are merged into one target language tree e^f if $q = u$ and $C_{e^{g,h}} = C_{e^g} \cup C_{e^h}$, resulting in a constructed grammar rule $f \notin G$ defined by the tuple $\langle d^f, e^f, A^f \rangle$, where $d^f = s_v$. The alignment of the constructed grammar rule is the union of the alignments of the grammar rules g and h: $A^f = A^g \cup A^h$.

As f is a constructed grammar rule, the absolute frequency of occurrence of the grammar rule $F(f) = 0$, which would result in $W(e^{g,h}) = 0$ in Eq. 17.1. In order to resolve this, the frequency of occurrence $F(f)$ is estimated according to Eq. 17.2.

$$F(f) = w(y_v) \times \frac{F(g)}{F(d^g)} \times \frac{F(h)}{F(d^h)} \tag{17.2}$$

where

- $w(y_v) = \sqrt[m]{\prod_{i=1}^{m} w(A_i^g)}$ is the weight of grammar rule g, which is the geometric mean of the weight of each individual occurrence of alignment A, as produced by the discriminative aligner described in 17.3.1;
- $F(g)$ is the frequency of occurrence of grammar rule g
- $F(d^g)$ is the frequency of occurrence of the source side d^g of grammar rule g
- $F(h)$ is the frequency of occurrence of grammar rule h
- $F(d^h)$ is the frequency of occurrence of the source side d^h of grammar rule g^h

Constructing grammar rules leads to overgeneration. As a filter the target language probability of such a rule is taken into account. This is estimated by multiplying the relative frequency of v_j in which c_i occurs as a child over all v_j's with the relative frequency of c_j occurring N times over c_j occuring any number of times, as shown in Eq. 17.3, which is applied recursively for every node $v_j \in V_e$ where V_e is the set of nodes in e^f.

$$P(e^f) = \prod_{j=1}^{m} \prod_{i=1}^{n} \frac{F(\#(c_i|v_j) \geq 1)}{F(v_j)} \times \frac{F(\#(c_i|v_j) = N)}{\sum_{r=1}^{n} F(\#(c_i|v_j) = r)} \tag{17.3}$$

where

$\#(c_i|v_j)$ is the number of children of v_j with the same label as c_i
N is the number of times the label c_i occurs in the constructed rule

The new weight $w(e^f)$ is calculated according to Eq. 17.4.

$$w(e^f) = \sqrt[cp]{F(f) \times P(e^f)} \tag{17.4}$$

where

cp is the construction penalty: $0 \leq cp \leq 1$.

When constructing a horizontally complete subtree fails, a grammar rule is constructed by translating each child separately.

17.5 Generation

The main task of the *target language generator* is to determine word order, as the packed forest contains unordered trees. An additional task of the target language model is to provide additional information concerning lexical selection, similar to the language model in phrase-based SMT [23].

The target language generator has been described in detail in [47], but the system has been generalised and improved and was adapted to work with weighted packed forests as input.

For every node in the forest, the surface order of its children needs to be determined. For instance, when translating *"een wettelijke reden"* into English, the bag $NP\langle JJ(legal), DT(a), NN(reason)\rangle$ represents the surface order of all permutations of these elements.

A large monolingual treebank is searched for an `NP` with an occurrence of these three elements, and in what order they occur most, using the relative frequency of each permutation as a weight. If none of the permutations are found, the system backs off to a more abstract level, only looking for the bag $NP\langle JJ, DT, NN\rangle$ without lexical information, for which there is most likely a match in the treebank.

When still not finding a match, all permutations are generated with an equal weight, and a penalty is applied for the distance between the source language word order and the target language word order to avoid generating too many solutions with exactly the same weight. This is related to the notion of distortion in IBM model 3 in [5].

In the example bag, there are two types of information for each child: the part-of-speech and the word token, but as already pointed out in Sect. 17.2 dependency information and lemmas are also at our disposal.

All different information sources (token, lemma, part-of-speech, and dependency relation) have been investigated with a back-off from most concrete (token + lemma + part-of-speech + dependency relation) to most abstract (part-of-speech).

The functionality of the generator is similar to the one described in [17], but relative frequency of occurrence is used instead of n-grams of dependencies. As shown in [47] this approach outperforms SRILM 3-g models [41] for word ordering. [51] uses *feature templates* for translation candidate reranking, but these can have a higher depth and complexity than the context-free rules used here.

Large monolingual target language treebanks have been built by using the target sides of the parallel corpora and adding the British National Corpus (BNC)[8].

17.6 Evaluation

We evaluated translation quality from Dutch to English on a test set of 500 sentences with three reference translations, using BLEU [34], NIST [9] and translation edit rate (TER) [40], as shown in Table 17.1.

We show the effect of adding data, by presenting the results when using the Europarl (EP) corpus, and when adding the OPUS corpus, the DGT corpus, and the private translation memory (transmem), and we show the effect of adding a dictionary of + 100,000 words, taken from the METIS Dutch English translation engine [6, 46]. This dictionary is only used for words where the grammar does not cover a translation.

[8] http://www.natcorp.ox.ac.uk/

Table 17.1 Evaluation of the Dutch-English engine

	Without dictionary			With dictionary		
Training data	BLEU	NIST	TER	BLEU	NIST	TER
EP	25.48	7.36	61.12	25.75	7.43	60.38
EP+OPUS	26.23	7.40	61.63	26.46	7.44	61.42
EP+OPUS+DGT	24.10	6.59	64.08	25.82	7.28	61.83
EP+OPUS+transmem	**29.12**	**7.68**	**60.04**	**29.33**	**7.71**	59.98
EP+OPUS+DGT+transmem	28.50	7.59	60.22	29.31	**7.71**	**59.47**

These results show that the best scoring condition is trained on all the data apart from DGT, which seems to deteriorate performance. Adding the dictionary is beneficial under all conditions. Error analysis shows that the system often fails when using the back-off models, whereas it seems to function properly when horizontally complete subtrees are found.

Comparing the results with Moses[9] [24] shows that there is a long way to go for our syntax-based approach until we par with phrase-based SMT. The difference in score is partly due to remaining bugs in the PaCo-MT system which cause no output in 2.6 % of the cases. Another reason could be the fact that automated metrics like BLEU are known to favour phrase-based SMT systems. Nevertheless, the PaCo-MT system has not yet reached its full maturity and there are several ways to improve the approach, as discussed in Sect. 17.7.

17.7 Conclusions and Future Work

With the research presented in this paper we wanted to investigate an alternative approach towards MT, not using *n*-grams or any other techniques from phrase-based SMT systems.[10]

A detailed error analysis and comparison between the different conditions will reveal what can be done to improve the system. Different parameters in alignment can result in more useful information from the same set of data. Different approaches to grammar induction could also improve the system, as grammar induction is now limited to horizontally complete subtrees. STSGs allow more complex grammar rules including horizontally incomplete subtrees. Another improvement can be expected from working on the back-off strategy in the transducer, such as the real time construction of new grammar rules on the basis of partial grammar rules.

[9]This phrase-based SMT system was trained on the same test set with the same training data, using 5-g without minimum error rate training scored 41.74, 43.30, 44.46, 49.61 and 49.98 BLEU respectively.

[10]Apart from word alignment.

The system could be converted into a syntactic translation aid, by only taking the decisions of which it is confident, backing off to human decisions in cases of data sparsity. It remains to be tested whether this approach would be useful.

Further investigation of the induced grammar could lead to a reduction in grammar rules, by implementing a default inheritance hierarchy, similar to [13], speeding up the system, without having any negative effects on the output.

The current results of our system are in our opinion not sufficient to reject nor accept a syntax-based approach towards MT as an alternative for phrase-based SMT, as, quoting Kevin Knight *"the devil is in the details"*.[11]

References

1. Aho, A., Ullman, J.: Syntax directed translations and the pushdown assembler. J. Comput. Syst. Sci. **3**, 37–56 (1969)
2. Bangalore, S., Joshi, A. (eds.): Supertagging. MIT, Cambridge, Massachusetts (2010)
3. Bod, R.: A Computational Model of Language Performance: Data-Oriented Parsing. In: Proceedings of the 15th International Conference on Computational Linguistics (COLING), Nantes, France, pp. 855–856 (1992)
4. Boitet, C., Tomokiyo, M.: Ambiguities and ambiguity labelling: towards ambiguity data bases. In: R. Mitkov, N. Nicolov (eds.) Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP), Tsigov Chark, Bulgaria (1995)
5. Brown, P., Cocke, F., Della Pietra, S., V.J., D.P., Jelinek, F., Lafferty, J., Mercer, R., Roossin, P.: A statistical approach to machine translation. Comput. Linguist. **16**(2), 79–85 (1990)
6. Carl, M., Melero, M., Badia, T., Vandeghinste, V., Dirix, P., Schuurman, I., Markantonatou, S., Sofianopoulos, S., Vassiliou, M., Yannoutsou, O.: METIS-II: low resources machine translation : background, implementation, results, and potentials. Mach. Trans. **22**(1), 67–99 (2008)
7. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd Annual Meeting of the ACL, Ann Arbor, US, pp. 263–270. ACL (2005)
8. Chiang, D.: An introduction to synchronous grammars. COLING/ACL Tutorial, Sydney, Australia (2006)
9. Doddington, G.: Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In: Proceedings of the Human Language Technology Conference (HLT), San Diego, USA, pp. 128–132 (2002)
10. Eisner, J.: Learning non-isomorphic tree mappings for machine translation. In: Proceedings of the 41st Annual Meeting of the ACL, Sapporo, Japan, pp. 205–208. ACL (2003)
11. Fox, H.: Phrasal cohesion and statistical machine translation. In: Proceedings of the 2002 conference on Empirical Methods in Natural Language Processing, Philadelphia, USA, pp. 304–311 (2002)
12. Galley, M., Hopkins, M., Knight, K., Marcu, D.: What's in a translation rule? In: Proceedings of the HLT Conference of the North American Chapter of the ACL (NAACL), Boston, USA, pp. 273–280 (2004)

[11]Comment of Kevin Knight on the question why syntax-based MT does not consistently perform better or worse than phrase-based SMT, at the 2012 workshop "More Structure for Better Statistical Machine Translation?" held in Amsterdam.

13. Gazdar, G., Klein, E., Pullum, G., Sag, I.: Generalized Phrase Structure Grammar. Blackwell, Oxford, UK (1985)
14. Graham, Y.: Sulis: An Open Source Transfer Decoder for Deep Syntactic Statistical Machine Translation. Prague Bull. Math. Linguist. **93**, 17–26 (2010)
15. Graham, Y., van Genabith, J.: Deep Syntax Language Models and Statistical Machine Translation. In: Proceedings of the 4th Workshop on Syntax and Structure in Statistical Translation (SSST-4), Beijing, China, pp. 118–126 (2010)
16. Graham, Y., van Genabith, J.: Factor templates for factored machine translation models. In: Proceedings of the 7th International Workshop on Spoken Language Translation (IWSLT), Paris, France (2010)
17. Guo, Y., van Genabith, J., Wang, H.: Dependency-based N-gram Models for General Purpose Sentence Realisation. In: Proceedings of the 22nd International Conference on Computational Linguistics (COLING), Manchester, UK, pp. 297–304 (2008)
18. Hassan, H., Sima'an, K., Way, A.: Supertagged phrase-based statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, pp. 288–295 (2007)
19. Hearne, M., Tinsley, J., Zhechev, V., Way., A.: Capturing Translational Divergences with a Statistical Tree-to-Tree Aligner. In: Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI), Skvde, Sweden (2007)
20. Hearne, M., Way, A.: Seeing the wood for the trees. Data-Oriented Translation. In: Proceedings of MT Summit IX, New Orleans, US (2003)
21. Klein, D., Manning, C.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the ACL, Sapporo, Japan, pp. 423–430. ACL (2003)
22. Koehn, P.: Europarl: a parallel corpus for statistical machine translation. In: Proceedings of MT Summit X, Phuket, Thailand, pp. 79–97. IAMT (2005)
23. Koehn, P.: Statistical Machine Translation. Cambridge (2010)
24. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., D., D., Bojar, O., Constantin, A., Herbst, E.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), Prague, Czech Republic, pp. 177–180 (2007)
25. Kotzé, G.: Improving syntactic tree alignment through rule-based error correction. In: Proceedings of ESSLLI 2011 Student Session, Ljubljana, Slovenia, pp. 122–127 (2011)
26. Kotzé, G.: Rule-induced correction of aligned parallel treebanks. In: Proceedings of Corpus Linguistics, Saint Petersburg, Russia (2011)
27. Lavie, A.: Stat-xfer: A general serach-based syntax-driven framework for machine translation. In: Proceedings of thr 9th International Conference on Intelligent Text Processing and Computational Linguistics, Haifa, Israel, pp. 362–375 (2008)
28. Lewis, P., Stearns, R.: Syntax-directed transduction. J. ACM **15**, 465–488 (1968)
29. Lundborg, J., Marek, T., Mettler, M., Volk, M.: Using the Stockholm TreeAligner. In: Proceedings of the 6th Workshop on Treebanks and Linguistic Theories, Bergen, Norway, pp. 73–78 (2007)
30. Marcu, D., Wang, W., Echihabi, A., Knight, K.: SPMT: statistical machine translation with syntactified target language phrases. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Sydney, Australia (2006)
31. de Marneffe, M., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC), Genoa, Italy (2006)
32. van Noord, G.: At last parsing is now operational. In: Proceedings of Traitement Automatique des Langues Naturelles (TALN), Leuven, Belgium, pp. 20–42 (2006)
33. Och, F., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29**(1), 19–51 (2003)
34. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

35. Poutsma, A.: Machine Translation with Tree-DOP. In: R. Bod, R. Scha, K. Sima'an (eds.) Data-Oriented Parsing, chap. 18, pp. 339–358. CSLI, Stanford, US (2003)
36. Probst, K., Levin, L., Peterson, E., Lavie, A., Carbonel, J.: MT for Minority Languages Using Elicitation-Based Learning of Syntactic Transfer Rules. Mach. Trans. **17**(4), 245–270 (2002)
37. Riezler, S., Maxwell III, J.: Grammatical Machine Translation. In: Proceedings of the HLT Conference of the North American Chapter of the ACL (NAACL), New York, USA, pp. 248–255 (2006)
38. Schabes, Y.: Mathematical and Computational Aspects of Lexicalized Grammars. Ph.D. thesis, University of Pennsylvania, (1990)
39. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK (1994)
40. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas (2006)
41. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing, Denver, USA (2002)
42. Tiedemann, J.: News from OPUS – a collection of multilingual parallel corpora with Tools and Interfaces. In: Proceedings of Recent Advances in Natural Language Processing (RANLP-2009), Borovets, Bulgaria, pp. 237–248 (2009)
43. Tiedemann, J.: Lingua-align: an experimental toolbox for automatic tree-to-tree alignment. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'2010), Valetta, Malta (2010)
44. Tiedemann, J., Kotzé, G.: A discriminative approach to tree alignment. In: Proceedings of Recent Advances in Natural Language Processing (RANLP-2009), Borovets, Bulgaria (2009)
45. Vandeghinste, V.: Removing the distinction between a translation memory, a bilingual dictionary and a parallel corpus. In: Proceedings of Trannslation and the Computer 29, ASLIB, London, UK (2007)
46. Vandeghinste, V.: A Hybrid Modular Machine Translation System. LoRe-MT: Low Resources Machine Translation. Ph.D. thesis, K.U. Leuven, Leuven, Belgium (2008)
47. Vandeghinste, V.: Tree-based target language modeling. In: Proceedings of the 13nd International Conference of the European Association for Machine Translation (EAMT-2009), Barcelona, Spain (2009)
48. Vandeghinste, V., Martens, S.: Top-down transfer in example-based MT. In: Proceedings of the 3rd Workshop on Example-based Machine Translation, Dublin, Ireland, pp. 69–76 (2009)
49. Vandeghinste, V., Martens, S.: Bottom-up transfer in example-based machine translation. In: Proceedings of the 14th International Conference of the European Association for Machine Translation (EAMT-2010), Saint-Raphal, France (2010)
50. Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., Nagy, V.: Parallel corpora for medium density languages. In: Proceedings of Recent Advances in Natural Language Processing (RANLP-2005), Borovets, Bulgaria, pp. 590–596 (2005)
51. Velldal, E., Oepen, S.: Statistical ranking in tactical generation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Sydney, Australia (2006)
52. Wang, W., May, J., Knight, K., Marcu, D.: Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. Comput. Linguist. **36**(2), 247–277 (2010)
53. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Comput. Linguist. **23**, 377–404 (1997)
54. Yamada, K., Knight, K.: A syntax-based statistical translation model. In: Proceedings of the 39th Annual Meeting of the ACL, Toulouse, France, pp. 523–530. ACL (2001)
55. Zollmann, A., Venugopal, A.: Syntax augmented machine translation via chart parsing. In: Proceedings of the Workshop on Statistical Machine Translation, New York, USA, pp. 138–141 (2006)

Part IV
HLT Application Related Papers

Chapter 18
Development and Integration of Speech Technology into COurseware for Language Learning: The DISCO Project

Helmer Strik, Joost van Doremalen, Jozef Colpaert, and Catia Cucchiarini

18.1 Introduction

Language learners seem to learn best in one-on-one interactive learning situations in which they receive optimal corrective feedback. The two sigma benefit demonstrated by Bloom [1] has provided further support for the advantages of one-on-one tutoring relative to classroom instruction. However, one-on-one tutoring by trained language instructors is costly and therefore not feasible for the majority of language learners. In the classroom, providing individual corrective feedback is not always possible, mainly due to lack of time. This particularly applies to oral proficiency, where corrective feedback has to be provided immediately after the utterance has been spoken, thus making it even more difficult to provide sufficient practice in the classroom.

The emergence of Computer Assisted Language Learning (CALL) systems that make use of Automatic Speech Recognition (ASR) seems to offer new perspectives for training oral proficiency. These systems can potentially offer extra learning time and material, specific feedback on individual errors and the possibility to simulate realistic interaction in a private and stress-free environment. For pronunciation training, systems have been developed that either provide overall scores of pronunciation performance or try to diagnose specific pronunciation errors

H. Strik (✉) · J. van Doremalen · C. Cucchiarini
CLST, Radboud University, Erasmusplein 1, 6525 HT, Nijmegen, The Netherlands
e-mail: W.Strik@let.ru.nl; j.vandoremalen@let.ru.nl; c.cucchiarini@let.ru.nl

J. Colpaert
Linguapolis, University of Antwerpen, Antwerp, Belgium
e-mail: jozef.colpaert@ua.ac.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_18, © The Author(s) 2013

[12, 14–16, 18, 22]; commercial systems are e.g., marketed by Digital Publishing,[1] Auralog,[2] and Rosetta Stone.[3] However, the level of accuracy achieved in signaling pronunciation errors to the learners is not always satisfactory [16].

Research at the Radboud University of Nijmegen has shown that a properly designed ASR-based CALL system is capable of detecting pronunciation errors and of providing comprehensible corrective feedback on pronunciation with satisfactory levels of accuracy [3]. This system, called Dutch-CAPT (Computer Assisted Pronunciation Training), was designed to provide corrective feedback on a selected number of speech sounds that had appeared to be problematic for learners of Dutch from various L1 backgrounds [17]. The results showed that for the experimental group that had been using Dutch-CAPT for 4 weeks the reduction in the pronunciation errors addressed in the training system was significantly larger than in the control group [3]. These results are promising and show that it is possible to use speech technology in CALL applications to improve pronunciation.

We therefore decided to extend this approach to other aspects of speaking proficiency like morphology and syntax. So far there are no systems that are capable of automatically detecting morphology and syntax errors in speaking performance and provide feedback on them. A project proposal which aimed to achieve this was funded by the STEVIN programme: the DISCO project. At the moment of writing the DISCO project has not been completed yet. Therefore, in this chapter we report on the research that has been carried out so far.

In the remainder of this chapter we first describe the aim of the DISCO project. We then go on to briefly deal with materials and method with respect to system design and speech technology components. Subsequently, we present the results of the DISCO project that are currently available. We then discuss the DISCO results, we consider how DISCO has contributed to the state of the art and present some future perspectives.

18.2 DISCO: Aim of the Project

The aim of the DISCO project was to develop a prototype of an ASR-based CALL application for Dutch as a second language (DL2). The application aims at optimising learning through interaction in realistic communication situations and providing intelligent feedback on important aspects of L2 speaking, viz. pronunciation, morphology, and syntax. The application should be able to detect and give feedback on errors that are made by DL2 learners.

L2 learners tend to make different morphologic and syntactic errors when they speak than when they write. It is generally acknowledged in the L2 literature

[1] http://www.digitalpublishing.de

[2] http://www.tellmemore.com/

[3] http://www.rosettastone.com/

that the fact that L2 learners are aware of certain grammatical rules (i.e. those concerning subject-verb concord of number, tenses for strong and weak verbs, and plural formation) does not automatically entail that they also manage to marshal this knowledge on line while speaking. In other words, in order to learn to speak properly, L2 learners need to practice speaking and to receive corrective feedback on their performance on line, both on pronunciation and on morphology and syntax. The ASR-based CALL system to be developed in the DISCO project was conceived to make this possible.

With respect to pronunciation, we aimed at the achievement of intelligibility, rather than accent-free pronunciation. As a consequence, the system was intended to target primarily those aspects that appear to be most problematic. In previous research [17] we gathered relevant information in this respect. In the DISCO project we wanted to extend the pronunciation component by providing feedback on more sounds and by improving the pronunciation error detection algorithms.

It is well-known that recognition of non-native speech is problematic. In the Dutch-CAPT system recognition of the utterances was successful because we severely restricted the exercises and thus the possible answers by the learners. Since DISCO also addresses morphology and syntax, the exercises have to be designed in such a way that L2 learners have some freedom in formulating their answers in order to show whether they are able to produce correct forms. So, the challenge in developing an ASR-based system for practicing oral proficiency consists in designing exercises that allow some freedom to the learners in producing answers, but that are predictable enough to be handled automatically by the speech technology modules.

In morphology and syntax we wanted to address errors that are known to cause problems in communication and that are known to be made at the low proficiency level (the so called A1/A2 proficiency level of the Common European Framework) that is required in national language citizenship examinations in the Netherlands (‘inburgeringsexamen’). For morphology this concerns (irregular) verb forms, noun plural formation; and for syntax it concerns word order, finite verb position, pronominal subject omission, and verb number and tense agreement.

The DISCO project is being carried out by a Dutch-Flemish team consisting of two academic partners, the Radboud University in Nijmegen (CLST and Radboud in’to Languages) and the University of Antwerp (Linguapolis), and the company Knowledge Concepts.

18.3 Material and Methods: Design

In this section we first describe the user interaction design and secondly the design of the speech technology modules utilised in the system.

18.3.1 User Interaction Design

The design model for the project was based on the engineering approach described in [2]. The design concepts for the application to be developed were derived from a thorough analysis of pedagogical and personal goals. While the pedagogical goals of this project were clearly formulated, for the elicitation of personal goals we needed to conduct a number of specific focus groups and in-depth interviews.

18.3.1.1 Interviews with DL2 Teachers and Experts

Exploratory in-depth interviews with DL2 teachers and experts were conducted. The results presented in this sub-section concern their opinions about DL2 learners.

Two types of DL2 learners were identified: those who want immediate corrective feedback on errors, and those who want to proceed with conversation training even if they make errors. Teachers also believed that our target group (highly-educated DL2 learners) would probably prefer immediate corrective feedback. To cater for both types of learners, the system could provide two types of feedback strategies and have the learners choose the one that suits them better through parameter setting.

The interviews also revealed that DL2 learners often want more opportunities to practice. A CALL system can provide these opportunities. DL2 learners feel uneasy at speaking Dutch because they are not completely familiar with the target language and culture. Therefore, it might be a good idea to provide some information about the target culture(s), so that learners can try to achieve intercultural competence.

18.3.1.2 Focus Group with DL2 Students

A focus group with nine DL2 learners revealed that DL2 learners preferred conversation simulation for building self-confidence over another traditional school-like approach. They also clearly preferred respect for their identity over explicit focus on integration.

DL2 learners often feel discouraged if they don't have sufficient knowledge of the topic of the conversation, for example politics, habits, etc. Furthermore, they want to feel respected for their courage to integrate in the target culture(s). The conversations may thus certainly deal with habits and practices of the target culture(s).

Also, learners feel frustrated because they cannot keep up with the pace of conversations in the target language. DL2 teachers and experts mentioned lack of exposure to L2 culture, but the participants did not complain about this lack, even if we explicitly asked them.

18.3.1.3 Conceptualisation

After an initial design based on a concept where the user was expected to make choices (communicative situation, pronunciation/morphology/syntax), we eventually decided to limit our general design space to closed response conversation simulation courseware and interactive participatory drama, a genre in which learners play an active role in a pre-programmed scenario by interacting with computerised characters or "agents".

The simulation of real-world conversation is closed and receptive in nature: students read prompts from the screen. However, at every turn, students pick the prompt of their choice, which grants them some amount of conversational freedom. The use of drama is beneficial for various reasons, (a) it "reduces inhibition, increases spontaneity, and enhances motivation, self-esteem and empathy" [13], (b) it casts language in a social context and (c) its notion implies a form of planning, scenario-writing and fixed roles, which is consistent with the limitations we set for the role of speech technology in DISCO [21].

This framework allows us to create an engaging and communicative CALL application that stimulates Dutch L2 (DL2) learners to produce speech and experience the social context of DL2. On the other hand, these choices are safe from a development perspective, and are appropriate for successfully deploying ASR while taking into account its limitations [10]. In order to make optimal choices with respect to important features of the system design, a number of preparatory studies was carried out in order to gain more insight into important features of system design such as feedback strategies, pedagogical and personal goals.

18.3.1.4 Prototyping

Pilot Study with DL2 Teachers

The current and the following pilot study were carried out by means of partial systems with limited functionality (e.g. no speech technology). The functions of the system that were not implemented such as playing prompts and giving feedback were simulated. For this pilot study, an internet application was used to present one conversation tree including graphics.

In general, DL2 teachers were positive about the possibilities offered by such a CALL system to practice pronunciation, morphology and syntax. Most of the comments dealt with how the exercises on morphology and syntax should be designed. The main conclusions were that different types of exercises probably require different approaches.

Pronunciation Exercises For pronunciation exercises, we decided that simply reading aloud sentences is a good modality for reliably detecting and correcting errors in pronunciation.

Morphology Exercises Regarding morphology, a multiple choice approach was recommended. For example, for personal and possessive pronouns: "Hoe gaat het met (jij/jou/jouw)?" ("How are (you/you/your)?") and for verb inflections: "Hoe (ga/gaat/gaan) het met jou?" ("How (are/is/to be) you?").

Syntax Exercises For syntax exercises, constituents can be presented in separate blocks in a randomised order. There shouldn't be too many of them (e.g. max. four) and some of these blocks could be fixed, such as the beginning and the end of the sentence. This can be made clear by using differently colored blocks.

Pilot Study with DL2 Students

A web-based prototype of the application was developed. A pronunciation teacher simulated the functions that were not yet implemented, e.g. by reading lines from the screen and providing feedback. The speech of the students was recorded, video recordings were made, and subsequently analyzed.

The pilot was carried out in Antwerp (five participants) and Nijmegen (four participants). The first research question concerned the feedback students prefer. Five out of nine respondents indicated a preference for immediate feedback, and four out of nine students responded that they did not know which feedback they preferred. The fact that no student wanted communicative (delayed) feedback confirms the hypothesis that highly-educated learners want to receive overt feedback with high frequency.

In exercises on morphology and syntax students first have to construct the grammatical form they want to utter. As a result, the cognitive load produced by these exercises is probably higher, which in turn may lead to a higher number of disfluencies and to speech recognition and error detection problems. A possible solution might be to ask students to first construct their answer on the screen by means of keyboard and mouse (textual interaction), and then utter these answers.

The average number of disfluencies per turn were measured by hand and we found that it was significantly lower in the cases with textual interactions. This shows that this procedure is useful to substantially reduce the number of disfluencies. However, CALL research does suggest that it is beneficial to maintain modalities, and not to use keyboard and mouse interaction in courseware that is essentially conversational in nature [13].

Furthermore, for some students it may not be necessary, or students may have a preference for not using it. Based on these results textual interaction could be included as an option and the output could be used to improve speech recognition and error detection.

Another important result from this pilot study is that the order of events was not always clear to students. Although the teacher that guided the experiment provided instructions that would normally be shown by the computer, students did things in the wrong order, acted ahead of time, spoke while carrying out the textual interaction, only uttered part of the prompts, or proceeded to the next item without speaking the utterance. The consequences for the design are that the interaction

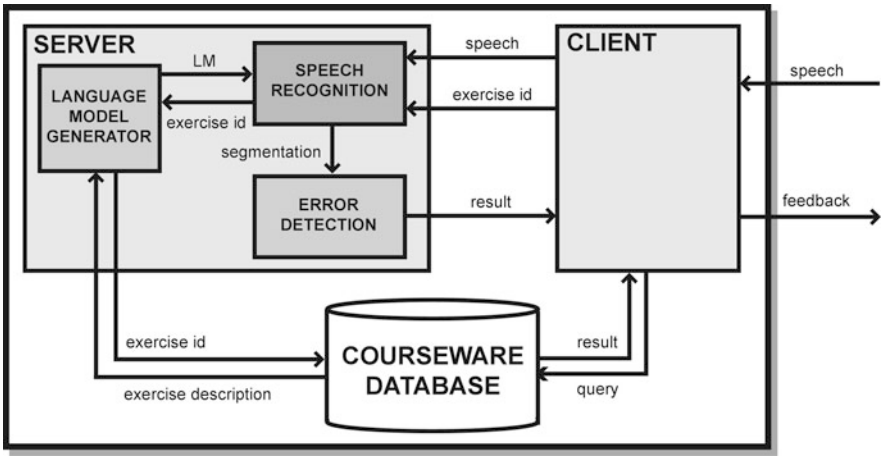


Fig. 18.1 Architecture of the DISCO system. More information is given in Sect. 18.3.2.1

sequence should be clearly structured and scaffolded, that instructions should be clear and concise, that a push-to-talk button should be used, and that students should be allowed to proceed to the next item if they have finished their task.

Finally, we also noticed that teachers, both in Nijmegen and in Antwerp, spontaneously provided non-verbal feedback during the conversation, and that students clearly responded to this kind of feedback. As CALL research also suggests [11], non-verbal feedback may be used complementarily to the verbal (overt or covert) feedback, and may be beneficial to student motivation and the learning effect. The virtual agents can provide this kind of feedback, e.g. by nodding or shaking their heads, smiling, frowning, etc.. However, we will need to be careful with showing this kind of feedback at all times, since it may become tiresome after a while. A random or intelligent random control for the non-verbal feedback may need to be implemented.

18.3.2 Speech Technology Design

18.3.2.1 System Architecture

Based on the exercises described in the previous section, we designed a system architecture which in principle is able to fulfill all the requirements stated during the courseware design phase (Fig. 18.1).

The system consists of three main components: the client, the server and the courseware database. The client will handle all the interaction with the user, such as recording the audio, showing the current exercise and appropriate feedback, as well as keeping track of the user's progress. The content of the courseware is stored in

the courseware database. The server is the component which processes the spoken utterances and detects errors.

In the DISCO application, the students' utterances have to be handled by the speech technology. For this purpose we employ a two-step procedure which is performed by the server: first it is determined what was said (content), and second how it was said (form). On the basis of the current exercise, the server generates a language model (language model generator) which is used by the speech recognition module to determine the sequence of words uttered by the student. If the speech recognition manages to do this, possible errors in the utterance are then detected by the error detection module. Finally, a representation of the spoken utterance, together with detected errors, is sent back to the client. The client then provides feedback to the learner.

The details of the design of the speech recognition and error detection modules are presented below.

18.3.2.2 Speech Recognition

For developing the speech recognition module the DISCO project has been able to profit from a previous STEVIN project, the SPRAAK project Chap. 6, which provided the speech recognition engine employed in DISCO.

During speech recognition, which is necessary to establish whether the learner produced an appropriate answer, the system should tolerate deviations in the way utterances are spoken. We call this step utterance selection. Exercises are designed such as to elicit constrained responses from the learner. For each exercise there is a specific list of predicted, correct and incorrect, responses. Incorrect responses are automatically generated using language technology tools based on the correct target responses.

Syntax Exercises In syntax exercises, three or four groups of words are presented on the screen. The task of the student is to speak these word groups in a syntactically correct order. For these exercises, language models are automatically generated by including all permutations of the word groups as paths in a finite state grammar (FSG). The task of the speech recogniser is to determine which of these paths in the FSG is the most likely one given the speech input from the student.

Morphology Exercises In morphology exercises, a whole sentence is presented on the screen, but for one word a multiple choice list containing alternatives for that word, typically around two to four, is presented. Here, the language models are generated in a similar fashion as in the syntax exercises. For the word that has to be chosen by the student, alternative paths are included in the FSG.

Pronunciation Exercises In pronunciation exercises, language models contain only one path: the target utterance. The reason for doing this recognition is explained below.

The sequence of words that is now selected does not always correspond exactly to what was actually spoken: the spoken utterance might not be present in the FSG, or even if it is present it might not be the one that is actually recognised. Since providing feedback on the wrong utterance is confusing, we try to avoid this as much as possible. To this end we automatically verify whether the recognised utterance was spoken using a so called confidence measure, which indicates how well the recognised word sequence reflects the spoken utterance. The confidence measure is compared to a predefined threshold to determine whether the utterance has to be accepted (confidence measure above the threshold) or rejected (below the threshold). This step is called utterance verification. When the utterance is accepted the learner gets feedback on the utterance, if it is rejected the learner might be asked to try again.

We conducted several experiments for optimising both utterance selection and utterance verification steps within the speech recognition module. These are described in Sect. 18.4.2.1.

18.3.2.3 Error Detection

After the speech recognition module has calculated the segmentation of the speech signal into words and phones, the error detection module detects errors on the levels of pronunciation, morphology and syntax. These types of error detection are explained below.

Pronunciation Exercises In previous studies [17] we investigated which pronunciation errors are made by learners of Dutch, and how these errors can be detected automatically. On the basis of three different databases, we drew up an inventory of frequent errors made by DL2 students [17]. Since Dutch has a rich vowel system, it is not surprising that many of the errors concern vowels. The distinction between tense and lax vowels, and the diphthongs appear to be problematic. Among the consonants the velar fricative /x/, a well-known shibboleth sound, and the glottal fricative /h/ seem to pose problems. For this reason we focused on detecting errors in the following phonemes: /i/, /ɪ/, /eː/, /ɛ/, /aː/, /ɑ/, /oː/, /ɔ/, /u/, /y/, /ʏ/, /ɛi/, /ʌu/, /øː/, /œy/, /x/, /ɦ/ and /ŋ/.

For pronunciation error detection, it has to be tested whether segments are realised correctly. We carried out multiple experiments to evaluate existing automatic methods for detecting these kinds of errors.

Syntax and Morphology Exercises While pronunciation error detection concerns detecting whether segments are correctly realised or not, syntactic and morphological error detection generally concerns detecting which words are correctly realised and whether they are in the right order. Because syntactically and morphologically incorrect responses are included in the list of predicted (correct and incorrect) responses, the output of the speech recognition module can thus be an incorrect utterance present in the predicted list and in this way errors can be detected.

Fig. 18.2 Screenshot of a morphology exercise in the DISCO system. The student gave the correct answer which is indicated by the *green block*. The functions of the four buttons on the right of the screen are (from left to right): start and stop recording speech input, listen to your own answer, listen to the prerecorded correct answer and proceed to the next prompt

18.4 Results

18.4.1 Design of the DISCO System

The results of the preparatory studies were taken into account in finalising the design of the DISCO system. The practice session starts with a relatively free conversation simulation, taking well into account what is (not) possible with speech technology: learners are given the opportunity to choose from a number of prompts at every turn (branching, decision tree, as shown in Fig. 18.2). Based on the errors they make in this conversation they are offered remedial exercises, which are very specific exercises with little freedom.

Feedback depends on individual learning preferences: the default feedback strategy is immediate corrective feedback, which is visually implemented through highlighting, and from an interaction perspective by putting the conversation on hold and focusing on the errors. Learners that wish to have more conversational freedom can choose to receive communicative recasts as feedback, which let the conversation go on while highlighting errors for a short period of time.

18.4.2 Speech Technology

18.4.2.1 Speech Recognition

For the purpose of developing the speech recognition module we used the JASMIN-CGN corpus (cf. Chap. 3, p. 43 to train and test experimental implementations. In a study in which we tested an experimental implementation of the speech recognition module, we showed that significant improvements relative to a baseline recognition system can be attained in several ways. The details of this experiment are described in [9].

The baseline system was a standard HMM-based speech recogniser with acoustic models trained on native speech. The language models were FSGs with about 30 to 40 parallel paths containing answers from non-native speakers to questions (from the JASMIN-CGN speech corpus). This baseline system had an utterance error rate UER of 28.9 %. The UER could be decreased to 22.4 % by retraining the acoustic phone models with non-native speech.

Furthermore, we found that filled pauses, which are very frequent in non-native speech [4], can be handled properly by including 'filled pause'-loops in the language model. Filled pauses are common in everyday spontaneous speech and generally do not hamper communication. Students are therefore allowed to produce (a limited number of) filled pauses. By using phone models trained on non-native speech and language models with filled pause loops, the UER of the speech recognition module in this task was reduced to 9.4 %.

As explained in Sect. 18.3.2.3, after the selection of the best matching utterance, the utterance verification step is needed to verify whether the selected response was indeed the utterance that was actually spoken by the learner. In [9] we presented and evaluated different methods for calculating confidence measures that are employed for this verification step.

The best results were obtained through a combination of acoustic likelihood ratios and phone duration features using a logistic regression model. The acoustic likelihood ratio indicates how well the acoustic features calculated from the speech match with the recognised utterance. Using only this feature the system has an equal error rate (EER) of 14.4 %. The phone duration features measure the number of extremely short (lower than the 5th percentile duration measured in a native speech database) and long (higher than the 95th percential duration) phones. By adding these features to the regression model the EER is decreased to 10 %.

18.4.2.2 Error Detection

In the current system design syntactical and morphological errors can already be detected after speech recognition, so no additional analysis is needed for these kinds of errors. However, for pronunciation errors such an analysis is required because these errors often concern substitutions of acoustically similar sounds. Therefore, considerable research efforts were made to improve the detection of pronunciation errors.

First, we conducted an experiment with artificial pronunciation errors in native speech [5]. We introduced substitutions of tense with lax vowels and vice versa, which is an error pattern frequently found in non-native speech. The results of this experiment show that discriminative training using Support Vector Machines (SVM's) based on acoustic features results in better pronunciation error classifiers than traditional acoustic likelihood ratios (LLR) (EER's of 13.9 % for SVM classifiers versus 18.9 % for LLR-based scores).

After having invested in improving the annotation of non-native read and spontaneous speech material in the JASMIN-CGN speech corpus, we first studied whether and how the error patterns of these two types of speech material differ in terms of phoneme errors [6]. We concluded that these two types of material indeed contained different phonemic error patterns, which partly depend on the influence of Dutch orthography [7].

Furthermore, we observed specific vocalic errors related to properties of the Dutch vowel system and orthography. We used this knowledge to develop a new type of pronunciation error classifier, which is designed to automatically capture specific error patterns using logistic regression models [7] and [8]. These classifiers performed better than acoustic LLR-based scores with average EERs of 28.8 % for the LLR-based scores and 22.1 % for the regression models).

18.4.3 System Implementation

We implemented the system architecture as depicted in Fig. 18.1. As stated in Sect. 18.3.2.1. the system has three main components: the client, the courseware database and the speech processing server. One of the advantages of separating client and server is that these components can be developed relatively independently, as long as the communication protocol is clearly defined. In most cases this might be the optimal set-up because different components will typically be developed by different experts, for example interaction designers, language teachers and speech technologists. The protocol was devised before developing the client and the server and it caters for both the transmission of audio and status messages (speech recogniser ready to receive speech, recognition started, recognition finished etc.). We chose to use one central server that can handle multiple clients because this is easy to maintain and update.

The client is implemented in Java using the AWT toolkit. The user-system interactions, the learners results, and the courseware, are stored in the relational MySQL courseware database. The speech processing server, which is the component which processes the spoken utterances and detects possible errors, is implemented in Python. The SPRAAK speech recogniser, implemented in C with an API in Python, is used in the speech recognition module. To handle multiple recognition requests a queueing system was implemented in which a constant number of recognisers is initialised. If all the recognisers in the queue recognise when a new recognition

request from a client comes in, this request is processed only after one of the recognisers has finished. This queueing method makes the system easily scalable.

Due to practical constraints, the speech recogniser's phone models are trained on native speech, the utterance verification is performed by only using an acoustic LLR measure and for pronunciation error detection we have also used acoustic LLR measures.

18.4.4 Evaluation

As mentioned above, various components of the system were evaluated at different stages in the project: the exercises, the speech recognition module, the error detection module, and finally the whole system as a preparation of the final evaluation. For the final evaluation of the whole system we chose an experimental design in which different groups of DL2 students at UA and Radboud into Languages use the system and fill in a questionnaire with which we can measure the students' satisfaction in working with the system. The student-system interactions are recorded. Experts then assess these recordings (the system prompts, student responses, system feedback, etc.) to study the interaction and especially the quality of the feedback on the level of pronunciation, morphology and syntax. At the moment of writing this evaluation is being conducted.

Given the evaluation design sketched above, we consider the project successful from a scientific point of view if the DL2 teachers agree that the system behaves in a way that makes it useful for the students, and if the students rate the system positively on its most important aspects.

18.5 Related Work and Contribution to the State of the Art

Within the framework of the DISCO project various resources have been developed. First of all a blue-print of the design and the speech technology modules for recognition (i.e. for selecting an utterance from the predicted list, and verifying the selected utterance) and for error detection (errors in pronunciation, morphology, and syntax). In addition: an inventory of errors at all these three levels, a prototype of the DISCO system with content, specifications for exercises and feedback strategies, and a list of predicted correct and incorrect utterances.

The fact that DISCO is being carried out within the STEVIN programme implies that its results, all the resources mentioned above, will become available for research and development through the Dutch Flemish Human Language Technology (HLT) Agency (TST-Centrale[4]).This makes it possible to reuse these resources

[4] www.tst-centrale.org

for conducting research and for developing specific applications for ASR-based language learning.

In addition, within DISCO research was conducted to optimise different aspects of the system. For instance, [9] presented research aimed at optimising automatic speech recognition for low-proficient non-native speakers, which is an essential element in DISCO. [5] addressed the automatic detection of pronunciation errors, while in [7] and [8] we described research on alternative automatic measures of pronunciation quality.

In [6] we studied possible differences in pronunciation error incidence in read and spontaneous non-native speech. Finally, research on automatic detection of syntactical errors in non-native utterances was reported on in [19] and [20].

Apart from the resources that become available during development of the system, additional resources can be generated by using the CALL system after it has been developed. Language learners can use it to practice oral skills and since the system has been designed and developed so as to log user-system interactions, these can be employed for research. The logbook can contain various information: what appeared on the screen, how the user responded, how long the user waited, what was done (speak an utterance, move the mouse and click on an item, use the keyboard, etc.), the feedback provided by the system, how the user reacted on this feedback (listen to example (or not), try again, ask for additional, e.g. meta-linguistic, feedback, etc.).

Finally, all the utterances spoken by the users can be recorded in such a way that it is possible to know exactly in which context the utterance was spoken, i.e. it can be related to all the information in the logbook mentioned above. An ASR-based CALL system like DISCO, can thus be used for acquiring additional non-native speech data, for extending already existing corpora like JASMIN-CGN, or for creating new ones. This could be done within the framework of already ongoing research without necessarily having to start corpus collection projects.

Such a corpus and the log-files can be useful for various purposes: for research on language acquisition and second language learning, studying the effect of various types of feedback, research on various aspects of man-machine interaction, and of course for developing new, improved CALL systems. Such a CALL system will also make it possible to create research conditions that were hitherto impossible, thus opening up possibilities for new lines of research.

For instance, at the moment a project is being carried out at the Radboud University of Nijmegen, which is aimed at studying the impact of corrective feedback on the acquisition of syntax in oral proficiency.[5] Within this project the availability of an ASR-based CALL system makes it possible to study how corrective feedback on oral skills is processed on-line, whether it leads to uptake in the short term and to actual acquisition in the long term. This has several advantages compared to other studies that were necessarily limited to investigating interaction in the written modality: the learner's oral production can be assessed on line,

[5] http://lands.let.kun.nl/~strik/research/FASOP.html

corrective feedback can be provided immediately under near-optimal conditions, all interactions between learner and system can be logged so that data on input, output and feedback are readily available for research.

18.6 Discussion and Conclusions

In the previous sections we have presented the various components of the DISCO system, how they have been developed, the results that have been obtained so far, and the resources that have been produced. The methodological design of the system has led to a software architecture that is sustainable and scalable, a straightforward interface that appeals to – and is accepted by – the users (by responding to their subconscious personal goals), a sophisticated linguistic-didactic functionality in terms of interaction sequences, feedback and monitoring, and an open database for further development of conversation trees. However, for a more complete and detailed appreciation of the whole system we will have to await the results of the final evaluation which is now being conducted.

In this paper we have also seen how important language resources are for developing CALL applications and how fortunate it was for DISCO to be able to use the JASMIN-CGN speech corpus (cf Chap. 3, p. 43) and the SPRAAK toolkit (cf Chap. 6, p. 95). In addition, we have underlined the potential of such applications for producing new valuable language resources which can in turn be used to develop new, improved CALL systems.

References

1. Bloom, B.: The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. Educ. Res. **13**(6), 4–16 (1984)
2. Colpaert, J.: Elicitation of language learners' personal goals as design concepts. Innov. Lang. Learn. Teach. **4**(3), 259–274 (2010)
3. Cucchiarini, C., Neri, A., Strik, H.: Oral proficiency training in dutch L2: the contribution of ASR-based corrective feedback. Speech Commun. **51**(10), 853–863 (2009)
4. Cucchiarini, C., van Doremalen, J., Strik, H.: Fluency in non-native read and spontaneous speech. In: Proceedings of DiSS-LPSS Joint Workshop 2010, Tokyo (2010)
5. van Doremalen, J., Cucchiarini, C., Strik, H.: Automatic detection of vowel pronunciation errors using multiple information sources. In: Proceedings of ASRU 2009, Merano, pp. 580–585 (2009)
6. van Doremalen, J., Cucchiarini, C., Strik, H.: Phoneme errors in read and spontaneous non-native speech: relevance for CAPT system development. In: Proceedings of the SLaTE-2010 workshop, Tokyo (2010a)

7. van Doremalen, J., Cucchiarini, C., Strik, H.: Using non-native error patterns to improve pronunciation verification. In: Proceedings of Interspeech, Tokyo, pp.590–593 (2010b)
8. van Doremalen, J., Cucchiarini, C., Strik, H.: Automatic pronunciation error detection in non-native speech. submitted to J. Acoust. Soc. Am.
9. van Doremalen, J., Cucchiarini, C., Strik, H.: Optimizing automatic speech recognition for low-proficient non-native speakers. EURASIP J. Audio Speech Music Process. (2010d), http://asmp.eurasipjournals.com/content/2010/1/973954
10. van Doremalen, J., Cucchiarini, C., Strik, H.: Automatic speech recognition in CALL systems: the essential role of adaptation. Commun. Comput. Inf. Science, **126**, 56–69 (2011). Springer
11. Engwall, O., Bälter, O.: Pronunciation feedback from real and virtual language teachers. J. Comput. Assist. Lang. Learn. **20**(3), 235–262
12. Eskenazi, M.: Using automatic speech processing for foreign language pronunciation tutoring: Some issues and a prototype. Lang. Learn. Technol. **2**, 62–76 (1999)
13. Hubbard, P.: Interactive participatory dramas for language learning. Simul. Gaming **33**, 210–216 (2002)
14. Kim, Y., Franco, H., Neumeyer, L.: Automatic pronunciation scoring of specific phone segments for language instruction. In: Proceedings of Eurospeech, Rhodes, pp. 645–648 (1997)
15. Mak, B., Siu, M., Ng, M., Tam, Y.-C., Chan, Y.-C., Chan, K.-W.: PLASER: pronunciation learning via automatic speech recognition., In: Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications using Natural Language Processing, Edmonton, pp. 23–29 (2003)
16. Menzel, W., Herron, D., Bonaventura, P., Morton, R.: Automatic detection and correction of non-native English pronunciations. In: Proceedings of InSTILL, Dundee, pp. 49–56 (2000)
17. Neri, A., Cucchiarini, C., Strik, H.: Selecting segmental errors in L2 Dutch for optimal pronunciation training. Int. Rev. Appl. Linguist. **44**, 357–404 (2006)
18. Precoda, K., Halverson, C.A., Franco, H.: Effects of speech recognition-based pronunciation feedback on second-language pronunciation ability. In: Proceedings of InSTILL, Dundee, pp. 102–105 (2000)
19. Strik, H., van de Loo, J., van Doremalen, J., Cucchiarini, C.: Practicing Syntax in spoken interaction: automatic detection of syntactic errors in non-native utterances. In: Proceedings of the SLaTE-2010 workshop, Tokyo (2010)
20. Strik, H., van Doremalen, J., van de Loo, J., Cucchiarini, C.: Improving ASR processing of ungrammatical utterances through grammatical error modeling. In: Proceedings of the SLaTE-2010 workshop, Venice (2011)
21. Strik, H., Cornillie, F., Colpaert, J., van Doremalen, J., Cucchiarini, C. (2009) Developing a CALL system for practicing oral proficiency: how to design for speech technology, pedagogy and learners. In: Proceedings of the SLaTE-2009 workshop, Warwickshire (2011)
22. Witt, S.M.: Use of speech recognition in computer-assisted language learning. Doctoral Dissertation, Department of Engineering, University of Cambridge, Cambridge (1999)

Chapter 19
Question Answering of Informative Web Pages: How Summarisation Technology Helps

Jan De Belder, Daniël de Kok, Gertjan van Noord, Fabrice Nauze, Leonoor van der Beek, and Marie-Francine Moens

19.1 Introduction

The DAISY (Dutch lAnguage Investigation of Summarisation technology) project started from a practical problem. Many companies maintain a large website with informative content. The users of such a website (e.g., clients of the company, business partners) want to quickly find the information that is relevant for their information question without getting lost when navigating the company's website, and want immediately to be directed to the right part of information when typing an information need. Summarisation of the informative Web texts will help in finding the correct answer to the information need. Summarised and rhetorically classified segments of the Web page will help to automatically map a user's question with the relevant information on the page.

DAISY is joint work of teams of the Katholieke Universiteit Leuven, the Rijksuniversiteit Groningen and the company RightNow (formerly Q-go). The aim of DAISY is to develop and evaluate essential technology for automatic summarisation of Dutch informative texts. Innovative algorithms for Web page segmentation, rhetorical classification of page's segments, sentence compression and generation of well-formed Dutch text have been developed. In addition, a proof-of-concept demonstrator is being developed in collaboration with the company RightNow.

J. De Belder (✉) · M.-F. Moens
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001, Heverlee, Belgium
e-mail: jan.debelder@cs.kuleuven.be; Sien.Moens@cs.kuleuven.be

G. van Noord · D. de Kok
University of Groningen, Groningen, The Netherlands
e-mail: g.j.m.van.noord@rug.nl; d.j.a.de.kok@rug.nl

F. Nauze · L. van der Beek
RightNow, Amsterdam, The Netherlands
e-mail: fabrice.nauze@rightnow.com; leonoor.vanderbeek@rightnow.com

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_19,

The remainder of this chapter is organised as follows. In the next section, we define the problem at hand. In Sect. 19.3, we discuss the cleaning and the segmentation of Web pages, which then can be used as input for further processing, such as by the rhetorical classifier, discussed in Sect. 19.4. Then, we continue with the sentence compression and sentence generation aspects of the project, discussed in Sect. 19.5 and 19.6 respectively. Finally, we discuss the demonstrator, to show the different methods, and end with the conclusion in Sect. 19.8.

19.2 Problem Definition

The general aim of the project is to develop and implement essential methods and supporting algorithms for summarisation of informative texts written in Dutch, and apply and evaluate them with texts in the financial and social security domain that are currently posted on the World Wide Web.

More specifically, the aim is to develop novel and robust technologies for (1) Segmentation and salience detection of content; (2) Single-sentence compression and sentence generation; (3) Rhetorical classification of informative text. For testing and evaluation purposes a demonstrator is being built that generates complementary types of summary information: (1) A headline type summary of a single text or text segment; (2) A short textual summary composed of compressed sentences; (3) Metadata that describes the rhetorical role (e.g., procedure, definition) of the text or text segment of which the summary is made.

For example, take the following text fragment:

```
''SNS Bank heeft maatregelen getroffen voor veilig Internet Bankieren''
 (SNS Bank has taken measures to perform bank transactions in a safe way).
```

In the context of the discourse, the sentence can be reduced to

```
''Maatregelen voor veilig Internet Bankieren''
(Measures to perform bank transactions in a safe way).
```

Also, detected rhetorical roles can be attached as meta-data to texts and their summaries. For example:

```
Example of a procedure ''Verzenden met EasyStamp'' (Send with EasyStamp)

''selecteer het adres of typ postcode en huisnummer in
kies het gewicht van het poststuk
selecteer een envelop of etiket (veel soorten en maten zijn al gedefinieerd)
kies eventueel voor een logo of afbeelding die u mee wilt printen
druk op de printknop''

(select the address or type postcode and house number
choose the weight of the mail piece
select an envelope or label (many types and sizes are defined)
choose optionally a logo or image that you want to print
push the print button)
```

In the example above, the fragment would be classified as a procedure, one of the six types of rhetorical roles we detect.

Essential in summarisation is the reduction of content to its most essential (salient) constituents and the generation of a concise summary text or other representation (e.g., in the form of concepts) that can be easily and efficiently processed by humans or by machines. Research into automated summarisation of text goes back several decades, but becomes increasingly important when information has to be selected from or sought in large repositories of texts. For an overview on text summarisation we refer to [11, 28], the proceedings of the yearly Document Understanding Conference (DUC) (2000–2007), and the proceedings of their successor, i.e. the Text Analysis Conferences (TAC) (2008–2012) Many current summarisation systems just extract sentences that contain content terms occurring frequently in the text, that occur at certain discourse positions, that contain certain cue terms (e.g., "in conclusion"), or learn the importance of these and other sentence scoring features from a training set of example texts and their summaries. Hence, the state of the art in summarisation is still far from truly abstractive summarisation, fusion of information from different texts, generalising content, and producing fluent, sensible abstracts. We see a current research interest in moving beyond extraction towards compressing and generating suitable summary sentences (e.g., [3, 8, 21, 31]). However, research into summarisation of Dutch texts is limited (e.g.,[24]: summarisation of court decisions; [32, 33]: summarisation of speech; [23]: summarisation of magazine articles). Studies that integrate into the summarisation certain pragmatic communication roles of the content are new. Segmentation and summarisation of informative texts that contain, for instance, instructions and procedural content are seldom researched.

A text may fulfill various pragmatic communication roles. For instance, it may describe a procedure, inform about a fact, or give a definition. Such roles are signaled by certain rhetorical linguistic cues. It is important to type a text (segment) according to its rhetorical function, as such typing has been proven a valuable part in summarising textual content [9, 30]. In this project, we use rhetorical typing in order to answer certain types of questions with text to which a suitable role is attached in a question answering system. Rhetorical structures of texts have been studied by Mann and Thompson [19] and used for summarisation of expository texts by Marcu [20].

This research extends previous work on text segmentation. After studying the corpora, and based on the literature of discourse theories, we defined a limited, but important set of rhetorical roles that are characteristic of the informative texts (e.g., definition, procedure, rule, . . .). These also correspond to the types of questions with which people interrogate the finance and social security texts.

In Fig. 19.1, we schematically represent the different components, and how they interact with each other.

19.3 Cleaning and Segmentation of Web Pages

A first step in analysing text on Web pages consists of extracting the text from the Web page. For humans this is a trivial task: a single glance at a page suffices to distinguish the main content from the less important information. However, when

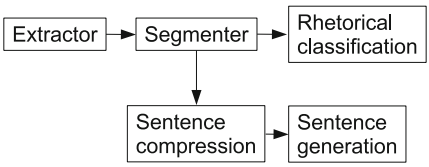


Fig. 19.1 Overview of different components and their interaction

only looking at the HTML code, it is often difficult to determine exactly where the main content starts and ends. Header, footers, menus, advertisements, ..., these are all elements that have to be taken into account, and dealt with properly.

The segmentation of Web pages goes a step further. For many Information Retrieval tasks a simple bag-of-words representation is sufficient, but here we also want the structural layout of the text. This means segmenting the text into sections, subsections, paragraphs,... and attaching the correct sections titles.

19.3.1 Content Extraction

The method we use for the extraction of the content performs only a very shallow analysis of the Web page. It does not depend on strong assumptions on the structure or content of the Web page and is fully language independent. The main idea behind the method is that a Web page has both content text and garbage text, but that the content texts tend to be continuous, long text with little structural markup, and that the garbage text tends to be short texts with a lot of structural markup. We make the following weak assumptions: The first assumption states that the text representing the content is separated from the garbage text with one or more markup tags. The second assumption states that no garbage text occurs in the main content, e.g. that the main content text is continuous (not taking into account the markup tags). The third and most important assumption states that the main content of the text contains less structural markup tags than the garbage text.

The method first locates a subset of markup tags that modify the structure of the Web page. These tags include, but are not limited to P, TABLE, BR, DIV, H1, H2 and LI tags. We ignore the tags that do not modify the structure of the Web page, such as B, A and FONT, and we also ignore data that is not content-related, such as JavaScripts, style definitions and HTML comments. We then transform the structured HTML page to a linear list of text strings $L = \{s_1, \ldots, s_n\}$. We parse the structure of the Web page using a robust HTML parser, that will, when presented with a not well-structured HTML page perform a best-effort parse. This parser visits every node in the HTML structure. If a node containing text is encountered, this text is added to the last text string in L. If a markup tag that modifies the structure of the Web page is encountered, L is extended with one empty string. We continue this process until the entire Web page is parsed.

We build a graphical representation of the array L in Fig. 19.2 where the x-axis represents the position of the array and the y-axis represents the length of the strings

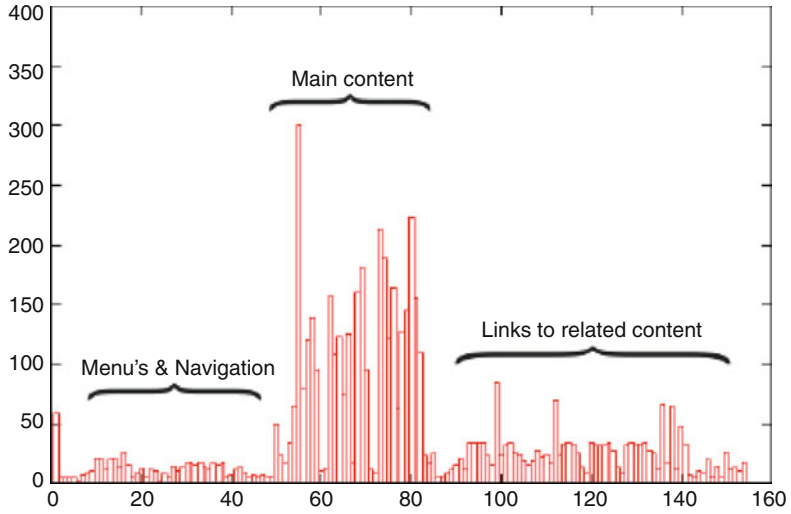


Fig. 19.2 Example plot of the document density

at the different positions. In a second step we analyse this graph to find the main content in the Web page. Typically, the main content for a Web page containing news articles is located in the region of L that has the highest density. We therefore convert the problem of extracting the main content of a Web page to the problem of selecting the highest density region of L, for which we have designed a simple but effective algorithm.

Although the method as a whole is very simple, it incorporates several interesting ideas. First of all, it does not depend on the structure of any particular Web site, but uses a notion of document density which can be expected to be universal for most Web sites containing news articles. Secondly, it does not depend in any way on the text and is thus fully language independent. Thirdly, it relies only on a limited amount of the HTML-markup, thus making allowances for dirty and non-well structured Web pages. For more details including the evaluation of the method on benchmarking corpora, see [1].

19.3.2 Segmentation

The nature of the Web pages in the corpora makes additional segmentation straightforward. HTML formatting tags present in the source files indicate text blocks (DIV), titles (H1, H2, H3), paragraphs (P), lists (UL, OL, DL), . . . providing strong clues about both the structure and the content (by looking at titles) of the text. An additional advantage is the generic nature of HTML, so the structure of any well-formed HTML page can be determined in a uniform fashion. Blocks of continuous text are further segmented into individual sentences.

The structure of the text provides cues for the rhetorical classification and the detection of the salient content in the document.

19.3.3 Corpora

The Web pages used in most of the experiments in the remainder of this chapter, were provided by RightNow. The company contacted several of its clients for the use of the data. Among those we have KLM,[1] UWV,[2] and SVB.[3]

19.4 Rhetorical Classification

For the purpose of better distinguishing parts of the Web pages, we classify the sentences as having a rhetorical role. We made a distinction between six relevant high level roles:

- **Definition** (DEF) A definition of a term, explaining its meaning.
- **Reference** (REF) A reference to another source for more background information, or a different source with the answer to the question (e.g. a phone number).
- **Information** (INF) An explanation or clarification about something (who, where, when, how much, which, why, ...-questions). In contrast to the commanding undertone that is present in Rules, there is a softer informative undertone that offers possibilities ('you can'), and not obligations ('you have to').
- **Meta-Information** (MIN) An explanation of why, which, and to what end the information is given on the page. It is information about the information that can be found on the page.
- **Procedure** (PRO) How a certain process is executed, or the different steps that need to be taken in order to complete something.
- **Rule** (RUL) A way one has to behave, i.e. an appointed or accepted norm, obligation, right or expectation, given in a commanding voice ('you have to', instead of 'you can').

These roles were developed by inspecting the corpora. They can be broken down further, e.g. a rule can be subdivided in a precondition and a postcondition. However, this could pose problems, as multiple of these more fine-grained roles can occur in a single sentence, which makes the classification task more difficult, and the training data will become sparse for some of the classes. We believe that these six high level roles are sufficient for practical use, and focus on them for now.

[1] http://www.klm.com/, the Royal Dutch Airlines

[2] http://www.uwv.nl/

[3] http://www.svb.nl/

19.4.1 Experiments

A set of 374 documents, selected randomly from the different corpora, was annotated with these roles. In total, we have 9,083 labelled sentences. The largest classes are "information" (32.8 %), "rule" (29.0 %) and "reference" (24.4 %). The other classes are smaller (8.8 % contains a "procedure", 3.5 % is a "definition", and 1.6 % is "meta-information"). It was however not always clear to which role a statement belonged. For instance, the difference between a piece of information and a rule are subject to personal judgement.

19.4.1.1 Baseline

As a baseline, we started by treating the problem as a multiclass text classification problem. We use unigrams and bigrams to represent the lexical aspect, and unigrams, bigrams, and trigrams of the Part-Of-Speech tags. We also include the POS tag of the first word as a feature, and several binary indicators for the presence of an imperative verb, a Multi-Word Unit, a wh-word, an auxiliary verb, possessives, and colons. The positional properties inform about the position in the paragraph, the depth in the hierarchy, and whether the sentence is actually a title. Finally, we include some statistics such as the number of words, the number of punctuation marks, and the average number of characters per word. We only kept the most significant features, according to a χ^2 test at $p < 0.05$.

We experimented with several algorithms, and found multinomial naive Bayes to be favourable compared to a maximum entropy classifier and a support vector machine. The results (of a ten-fold cross validation) show an accuracy of up to 70 % and a macro-averaged F1-measure up to 52.54 %. The rest of the results can be found in Table 19.1.

19.4.1.2 Improved Algorithm

Having exhausted the possible features and classification algorithms, we made use of additional information to improve results. Since the role of a sentence is dependent on the role of its surrounding sentences, and its position in the hierarchy, we try to find a globally optimal assignment for all the sentences in a document. We do so by building simple transition models, where we assign a probability of a label based on the label of the previous sentences, or the label of the sentence that is the parent in the hierarchy. Combining this with the probabilistic output of the multinomial naive Bayes classifier, we can find an assignment for all the sentences that maximises the probability of the document as a whole, by solving the corresponding optimisation problem with an Integer Linear Programming formulation.

Table 19.1 F_1 scores of the different methods. The column labelled *Baseline* indicates the baseline method, after applying feature selection. The *second column* indicates the sequential method, and the *third* the hierarchical method. The *last column* combines the two latter methods

Class	Baseline	Sequential	Hierarchic	Sequential + Hierarchic
DEF	46.39 %	55.81 %	54.4 %	58.06 %
DVW	82.78 %	84.03 %	83.34 %	84.4 %
INF	60.82 %	62.98 %	62.38 %	64.31 %
MIN	39.55 %	43.84 %	39.81 %	42.27 %
PRO	34.68 %	39.76 %	36.07 %	43.11 %
REG	51.0 %	53.81 %	51.57 %	53.69 %
Accuracy	70.73 %	74.06 %	72.55 %	75.21 %

By using the additional information given by the segmenter, and finding a globally optimal solution, we have obtained an average accuracy of 75 %, and a macro-averaged 57.64 % F1 score, thereby improving the baseline accuracy with 5 %. The complete results can be found in Table 19.1.

19.4.2 Conclusions and Future Work

In this component we have looked at assigning a rhetorical role to sentences in an informative document. This is a novel task, and there is no previous work with which we can compare. We initially treated the problem as a text classification problem. In order to improve the results, we combined this basic classifier with information from the previous component, i.e. the segmentation. Now a globally optimal assignment is found, and this led to improved results.

The obtained results are probably also the upper limit that can be reached without annotating more data. The rhetorical classification is a difficult task, as often it is hard to distinguish between the different roles.

Another possible line of research, is by using more data in an unsupervised setting. E.g. by taking the first sentences of each Wikipedia article, it is straightforward to obtain a corpus consisting of definitions. These can then be used to train a better classifier for definitions. A similar approach can be followed for procedures, e.g. by retrieving a set of instructional texts.

19.5 Sentence Compression

There exist a myriad of methods for the task of sentence compression, but the majority of these are hard to use in this case. The majority of methods learn how to compress sentences by learning from training data [13, 21, 31]. However,

manually creating training data is a time consuming and expensive task. Moreover, the few corpora that are available for Dutch, are from a completely different domain. Another aspect of this project that in a way limits the range of possibilities for sentence compression algorithms, is that Dutch is not a context free language, which means that we can't make use of the large number of methods for English that build on a Probabilistic Context Free Grammar (e.g. [8, 13]). Therefore, in this research we focused our attention on unsupervised methods, that are not too dependent on the output format of the parser.

We view sentence compression in a word removal setting. An advantage of such an approach is that sentence compression can be seen as a machine learning task, and many methods can be applied to it. In the literature we find, among others, a noisy channel model approach ([8, 13, 31]), a method based on integer linear programming [4] and a large margin online learning approach [21].

In this section we will define a uniform framework for compressing sentences using language models and optimisation. At the core of the algorithms lies the following problem: choose a subset of the words that maximise the probability of the compressed sentence in a language model. The major difference between the methods is the type of language model that is being used. Choosing this optimal subset of words can be done by solving an Integer Linear Programming problem. Below we sketch the broad ideas behind the methods.

19.5.1 Constrained Language Model Optimisation

We investigated three unsupervised methods, which we modelled in a similar fashion. Each of the methods share a similar problem formulation. They start from binary variables a_i for each word w_i in the sentence to be compressed. These variables can only have a value of 1 or 0, the former indicating that the word is included in the sentence, the latter indicating that w_i is not included in the compressed sentence.

With these a_i variables, and other variables depending on the model, we create a linear objective function. An optimal solution for the sentence compression problem is then found by finding an assignment for the variables that maximises the objective function. The difference between the methods lies in how they fill in this objective function.

19.5.1.1 Optimising an n-Gram Language Model

For a bigram language model, this roughly translates to assigning values of 0 or 1 to variables x_{ij}, each x_{ij} meaning that the bigram $w_i\ w_j$ is present in the compressed

sentence.[4] An optimal solution is then found by maximising:

$$\sum_{i=0} \sum_{j=i+1} x_{ij} P(w_j|w_i) \tag{19.1}$$

with $P(w_j|w_i)$ the probability that the word w_i is followed by the word w_j,

To ensure that the output is grammatically correct, and doesn't lose semantic information, an additional set of rules is applied, that are enforced in the form of constraints. These are based on a syntactic analysis of the sentence. The constraints state for example that when a verb is included in the compressed sentence, its subject and object also have to be included, and that a negation can not simply be removed from the word it modifies, etc.

19.5.1.2 Optimising a Dependency Language Model

A disadvantage of the previous method, is that the n-gram language model only finds fluent sentences locally. By using a language model defined over dependency trees, such as in [7], this problem is alleviated. In a dependency tree representation, words that are syntactically close together, are also close together in the model.

For each dependency ending in word a_i we have the parent word h_i, and l_i, the label of the dependency relation between a_i and h_i. The goal is then to maximise the following equation:

$$\sum_{i} a_i P(l_i|h_i) \tag{19.2}$$

with $P(l|h)$ the probability that a certain word h has a child with the dependency label l. This latter is estimated as $P(l|h) = \frac{Count(h,l)}{Count(h)}$, where the counts are obtained by parsing a sufficiently large corpus. E.g. most verbs have a subject, so $P(subj|have)$ will be high.

19.5.1.3 Optimising an Unfactored Dependency Language Model

A disadvantage of the method in [7], is that the probability of the children of a word are estimated separately ($P(l|h)$, the probability of word h having a child with label l between them). Our parsed corpus is however large enough, so that we can estimate $P(l_1, l_2, ..|h)$: the probability of a word having a set of children (e.g. the probability of a verb having a subject *and* an object, instead of the individual probabilities.

[4] In practice we use a trigram model, but for simplicity this is left out.

Additional Constraints

One of the most important functions of the constraints is to ensure that the problem is solved correctly. E.g. in equation 19.1, x_{13} can only have value 1 if $a_1 = 1, a_2 = 0, a_3 = 1$. Other possibilities with these constraints are stating that $\sum_{i=1}^{n} a_i \geq lowerbound$, to specify a minimum number of words.

Significance Model

To ensure that the compressed sentence contains the most important information, we modify the objective function, so that an additional 'bonus' is given for including a specific word. For each word, we calculate the importance with the following equation:

$$I(w_i) = \frac{l}{N} f_i log \frac{F_a}{F_i} \tag{19.3}$$

where f_i and F_i are the frequencies of word w_i in the document and a large corpus respectively, F_a the sum of all topic words in the corpus. l is based on the level of embedding of w_i: it is the number of clause constituents above w_i, with N being the deepest level in the sentence

19.5.2 Evaluation and Results

Using current evaluation measures, we can show that our unsupervised methods perform comparably with supervised methods. We not only evaluated on our own annotated small subset of the corpus, the results of which are available in Table 19.2, but also on existing corpora for sentence compression, of which our findings are in preparation. From Table 19.2, we can see the difference between the methods. Using only the n-gram language model and grammaticality constraints, the output is not so grammatical, but contains the most important information. When using language models based on the dependency trees, the output becomes more grammatical, but the score for the importance goes down, despite the longer sentences. The difference lies in the fact that the last two methods don't take into account the lexical items in the leaves of the dependency tree.

We also correlated different automatic evaluation measures with human judgement. Our results show that for Dutch, the evaluation measure based on the parse tree is the most correlated. This measure also takes the grammaticality into account, because if a sentence is ungrammatical, the parser will not be able to capture the dependencies between the words.

The annotation process of the informative texts was very enlightening. Annotators found it very difficult to compress sentences without the proper context. When

Table 19.2 Human ratings for each of the three methods, on a five point scale (5 being the highest score, 1 the lowest), grading the grammaticality and importance aspect. The *last column* indicates the average number of words in the compressed sentence

Method	Grammaticality	Importance	AvgNbWords
n-gram LM	2.60	3.23	12.1
Dependency LM	3.28	2.95	12.7
Joint dependency LM	3.67	2.62	12.9

faced with the complete text, this posed less of a problem, although it was still harder in comparison to texts containing a lot of redundant information.

In a practical setting, it is often faster to use a method with a language model based on dependency trees, rather than one with an n-gram language model. The disadvantage is that this yields a lower importance score, but this can be alleviated by using the Significance model. The trade-off between the two models then has to be estimated on a small validation set.

We refer the interested reader to other publications for more information [6].

19.6 Sentence Generation

19.6.1 Introduction

Since the sentence compression component deletes words, it is possible that the word order has to be changed. In order to reorganise the ordering of the words, we use a sentence realiser that, given the dependencies required in a sentence, arranges them for a fluent result.

Sentence realisers have been developed for various languages, including English and German. While the generation algorithms used in sentence realisers are very generic, the implementation of a realiser is quite specific to the grammar formalism and input representation. We developed a sentence realiser for the wide-coverage Alpino grammar and lexicon.

Alpino [25] is a parser for Dutch which includes an attribute-value grammar inspired by HPSG, a large lexicon, and a maximum entropy disambiguation component. Dependency structures are constructed by the grammar as the value of a dedicated attribute. These dependency structures constitute the output of the parser.

In generation, the grammar is used in the opposite direction: we start with a dependency structure, and use the grammar to construct one or more sentences which realise this dependency structure. Dependency structures that we use in generation contain less information than the dependency structures that are the output of parsing. For instance, information about word adjacency, separable particles and punctuation are removed. The user can also decide to underspecify

certain lexical information. We call such dependency structures *abstract dependency structures* [16].

In the general case, a given dependency structure can be realised by more than a single sentence. For instance, the sentence *Na de verkiezingen beklijfden de adviezen echter niet* (*After the elections the advises did, however, not persist.*) is mapped to a dependency structure which can also be realised by variants such as *Na de verkiezingen beklijfden de adviezen niet echter*, or *echter beklijfden na de verkiezingen de adviezen niet*. Therefore, a maximum entropy fluency ranker is part of the generator. The fluency ranker selects the most appropriate, 'fluent', sentence for a given dependency structure.

19.6.2 Chart Generation

In the Alpino generator, we use chart generation [12, 29]. This algorithm closely resembles bottom-up chart parsing, however guidance is provided by semantics rather than word adjacency.

For details of our sentence realiser, we refer to [16]. However, one interesting aspect of our realiser is that it implements top-down guidance differently than in previous work that we know of. Since the Alpino grammar is semantically monotonous [29], we could use a semantic filter that constrains generation. Such a filter excludes derivations where the semantics of the derivation do not subsume a part of the goal semantics. In our system, we use an even stronger approach: we instantiate each lexical item that represents a head in the dependency structure with its expected dependency structure. In this manner, it is not possible to construct partial derivations with dependency structures that do not subsume a part of the input dependency structure.

19.6.3 Fluency Ranking

A sentence realiser can often produce many different sentences for a given input. Although these sentences are normally grammatical, they are not all equally fluent. We developed a fluency ranker that attempts to select the most fluent sentence from a list of sentences.

Different statistical models have been proposed for fluency ranking in the past, such as n-gram language models, maximum entropy models, and support vector machines [34]. As [34] shows, maximum entropy models perform comparably to support vector machines for fluency ranking, while having a shorter training time. For this reason, we use a conditional maximum entropy model in our fluency ranker.

In our model probability of a realisation r given the dependency structure d is defined as:

Table 19.3 General Text Matcher scores for fluency ranking using various models

Model	GTM
Random	55.72
Trigram	67.66
Fluency	71.90

$$p(r|d) = \frac{1}{Z(d)} exp \sum_i \lambda_i f_i(d, r) \tag{19.4}$$

Where $f_i(f, r)$ is the value of feature f_i in the realisation r of d, λ_i the weight of that feature, and $Z(d)$ normalises over all realisations of the dependency structure d. Training the model gives a set of feature weights Λ that predicts the training data, but has as few other assumptions as possible.

Features are automatically extracted from the training data using feature templates. Our fluency ranker works with the following classes of features:

- *Word adjacency* is modelled using trigram language models of words and part-of-speech tags.
- *Shallow syntactic* features record rule applications and combinations of rule applications.
- *Syntactic* features describe various syntactic aspects of a realisation, such as fronting, depth and parallelism in conjunctions, and orderings in the middle-field.

19.6.4 Evaluation and Results

To evaluate the fluency ranker, we first trained a fluency ranking model using the *cdbl* part of the Eindhoven corpus[5] (7,154 sentences). Syntactic annotations are available from the Alpino Treebank[6] [2].

We then evaluated this model using a part of the Trouw newspaper of 2001 from the Twente Nieuwscorpus. [7] Syntactic annotations are part of Lassy [8] [26], part WR-P-P-H (2,267 sentences). For each pair of a sentence and dependency structure in the treebank, we consider the sentence to be the gold standard, and use the dependency structure as the input to the generator. We then use the General Text Matcher method [22] to compute the similarity of the most fluent realisation and the gold standard sentence.

Table 19.3 compares random selection, a word trigram model, and our fluency ranking model. As we can see in this table, our maximum entropy fluency ranking model outperforms both the random selection baseline and the word trigram model.

[5] http://www.inl.nl/corpora/eindhoven-corpus

[6] http://www.let.rug.nl/~vannoord/trees/

[7] http://hmi.ewi.utwente.nl/TwNC

[8] http://www.inl.nl/corpora/lassy-corpus

19.6.5 Further Research

In [14] we have compared various feature selection methods to reduce the size of the fluency ranking model and to get more insight into the discriminative features. We also developed the Reversible Stochastic Attribute-Value Grammar (RSAVG) formalism, that uses one model for both parse disambiguation and fluency ranking [17]. Subsequently, we have RSAVG to be truly reversible [15].

19.7 Proof-of-Concept Demonstrator

The developed technology is made publicly available through the demonstrator. This demonstrator is a Web-based interface that allows users to summarise sample texts, uploaded documents, or shorts texts, which the user enters in a textbox. A screenshot of the interface is shown in Fig. 19.3. For testing and evaluation purposes the demonstrator generates three complementary types of summary information: (1) A headline type summary of a single text or text segment; (2) A short textual summary composed of compressed sentences; (3) Metadata that describes the rhetorical role (e.g., procedure, definition) of the text or text segment of which the summary is made. The combination of the summaries and the metadata discriminate a text in a document base by the description of topics and the role of the text (segment) in the discourse.

Two lines of evaluation of the demonstrator will be pursued: an intrinsic and an extrinsic one. With intrinsic evaluation, the system's output is compared with humans' output and their congruence is computed. Extrinsic evaluation on the other hand, measures the quality as needed for other information tasks (e.g., filtering and retrieval).

We have performed an intrinsic evaluation with some common metrics from the Document Understanding Conference, namely 'Pyramid' [10] and 'Rouge' [18] and. When evaluating the demonstrator, the system output is compared against hand-made abstracts of the documents. Because of the problem of subjectivity of human summarisation, wherever possible three or more model summaries of the same text were collected. It is expected that good system-made summaries have a sufficient amount of congruence with at least one of the human-made summaries. The model summaries have been created by the company RightNow. Very often variant summaries made by different persons are available. In each step, both a baseline summary and the summaries generated by the demonstrator were compared with the model summary.

The effect of adding system-generated headline abstracts on retrieval will be measured. The summaries are used to assist the question answering system developed by RightNow in the search for precise answers to information queries posed by end-users. This extrinsic evaluation is very important. RightNow monitors the recall and precision of its question answering system. This data can be reused in

order to test whether recall and precision of the retrieval can be improved by adding automatically generated summaries to the system, or by replacing the hand-made abstracts with system summaries.

Currently, RightNow processes user questions based upon a lexical, syntactic and semantic analysis, which results in a formal representation. The application matches such representations against similar representations in a database. These database entries are the result of the linguistic analysis of "template questions". The template questions are created manually, and each question is associated with an answer, which may be a piece of content on the customer website, or a brief textual answer and a link to the relevant Web page.

We have manually crafted template questions and the short textual answers as one or more summarisations reflecting the gist of the target document, which is why we think that an applied summarisation system can replace or at least help a large part of the editorial procedure needed in the current setup. Furthermore, we hope to improve the retrieval by associating automatically created summaries to templates as an alternative for matching.

The obtained pyramid and rouge results of the DAISY summaries are comparable with what we see in the state-of-the-art literature of the DUC [27] and TAC [5] competitions organised by the National Institute of Standards and Technology (NIST) for the English language (where other types of texts such as news stories were summarised). Compared to uncompressed HTML text of the Web pages, there are few matching LCS (lowest common sub-sequences). This is mainly caused by three factors:

1. Even though the knowledge base content is linked to external Webpages, the match questions in the database try to model the way end-users formulate questions about the web content, it is not a model of the content itself.
2. The segmentation of the html text does not handle links and lists correctly.
3. The compression used for the evaluation is quite aggressive which has a great impact on matching sub-sequences.

This result was to be expected as content on informative Web pages is always important for a certain user and summarisation or compression is not always the correct answer to improve the matching in a question answering task.

We also evaluated whether summaries can replace match questions within the Intent Guide (IG) without loss in quality and whether the summaries can improve the current quality of the IG implementation. We run two experiments. In the first one only summaries were used for matching, in the second summaries were added to the model questions as match questions (the customer would therefore still need to create model questions – the questions displayed to the end-user as answer to his/her query).

The results of the first run show a match percentage of 30 % which is too low to replace the IG match question. However we do get new matches with the second test set which is positive. The second run shows that the addition of model questions improves the results greatly and that the summaries might be used as extra matching questions (improving the system but invisible to the end-user).

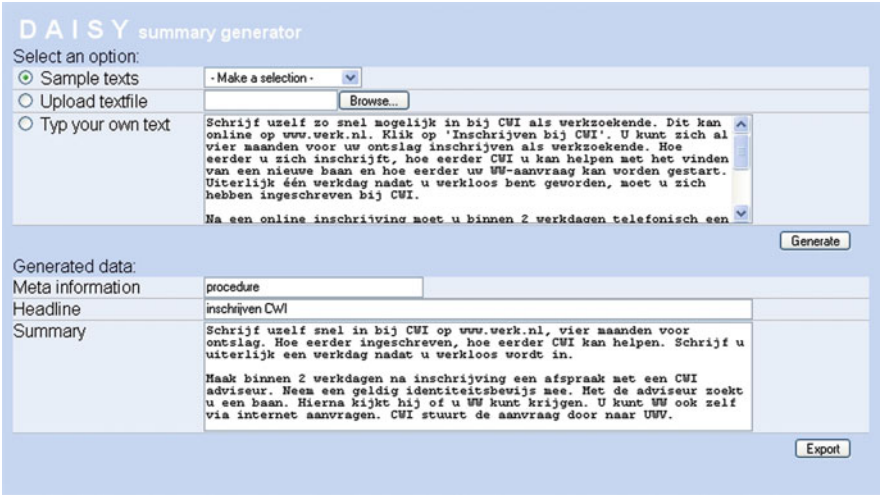


Fig. 19.3 Demonstrator interface

19.8 Conclusions

The novelty of our approach lies in (1) Classification of the rhetorical role of a text segment or sentence, using text automatically extracted from Dutch informative Web pages; (2) Improvements of current sentence compression technologies for Dutch texts; (3) Development of standard text generation technology for Dutch – integrated with the standard Dutch text analysis tools.

These tasks regard essential tasks in summarisation of informative content. The summarisation demonstrator can already be considered as an application. Because in informative Web pages any content is important in a certain circumstance for some user, it is difficult to compress this content. But, DAISY has contributed to generating additional paraphrases to the ones already used by RightNow for matching questions and answers.

References

1. Arias, J., Deschacht, K., Moens, M.F.: Content extraction from multilingual web pages. In: Proceedings of the 9th Dutch-Belgium Information Retrieval Workshop. University of Twente, Enschede, The Netherlands (2009)
2. van der Beek, L., Bouma, G., Malouf, R., van Noord, G.: The Alpino dependency treebank. In: Computational Linguistics in the Netherlands (CLIN), Groningen, The Netherlands (2002)

3. Clarke, J., Lapata, M.: Models for sentence compression: a comparison across domains, training requirements and evaluation measures. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 377–384. Association for Computational Linguistics, Sydney, Australia (2006)
4. Clarke, J., Lapata, M.: Global inference for sentence compression: an integer linear programming approach. J. Artif. Intell. Res. **31**(1), 399–429 (2008)
5. Dang, H.T., Owczarzak, K.: Overview of the TAC 2009 summarization track. In: Proceedings of the Second Text Analysis Conference (TAC2009), Gaithersburg, Maryland, USA (2009)
6. De Belder, J., Moens, M.F.: Integer linear programming for Dutch sentence compression. In: Computational Linguistics and Intelligent Text Processing, Iaşi, Romania, pp. 711–723 (2010)
7. Filippova, K., Strube, M.: Dependency tree based sentence compression. In: Proceedings of the Fifth International Natural Language Generation Conference, pp. 25–32. Association for Computational Linguistics, Columbus, Ohio, USA (2008)
8. Galley, M., McKeown, K.: Lexicalized Markov grammars for sentence compression. In: The Proceedings of NAACL/HLT, Rochester, NY, USA, pp. 180–187 (2007)
9. Hachey, B., Grover, C.: Automatic legal text summarisation: experiments with summary structuring. In: Proceedings of the 10th International Conference on Artificial intelligence and Law, pp. 75–84. ACM, Bologna, Italy (2005)
10. Harnly, A., Nenkova, A., Passonneau, R., Rambow, O.: Automation of summary evaluation by the Pyramid method. In: Proceedings of the Conference of Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria (2005)
11. Hovy, E., Marcu, D.: Automated text summarization. In: The Oxford Handbook of Computational Linguistics pp. 583–598. Oxford University Press (2005)
12. Kay, M.: Chart generation. In: Proceedings of the 34th Annual Meeting on ACL, pp. 200–204. ACL, Santa Cruz, California, USA (1996)
13. Knight, K., Marcu, D.: Statistics-based summarization-step one: sentence compression. In: Proceedings of the National Conference on Artificial Intelligence, pp. 703–710. MIT, Austin, Texas (2000)
14. de Kok, D.: Feature selection for fluency ranking. In: Proceedings of the 6th International Natural Language Generation Conference, pp. 155–163. Association for Computational Linguistics, Trim, Co. Meath, Ireland (2010)
15. de Kok, D.: Discriminative features in reversible stochastic attribute-value grammars. In: Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop, pp. 54–63. Association for Computational Linguistics, Edinburgh (2011). http://www.aclweb.org/anthology/W11-2708
16. de Kok, D., van Noord, G.: A sentence generator for Dutch. In: Proceedings of the 20th Computational Linguistics in the Netherlands conference (CLIN), Utrecht, The Netherlands (2010)
17. de Kok, D., Plank, B., van Noord, G.: Reversible stochastic attribute-value grammars. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, pp. 194–199. ACL, Portland, Oregon, USA (2011)
18. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain (2004)
19. Mann, W., Thompson, S.: Rhetorical structure theory: toward a functional theory of text organization. Text-Interdiscip. J. Study Discourse **8**(3), 243–281 (1988)
20. Marcu, D.: The theory and practice of discourse parsing and summarization. The MIT Press, Cambridge (2000)
21. McDonald, R.: Discriminative sentence compression with soft syntactic evidence. In: Proceedings of EACL, vol. 6, Trento, Italy, pp. 297–304 (2006)
22. Melamed, I.D., Green, R., Turian, J.: Precision and recall of machine translation. In: HLT-NAACL, Edmonton, Canada (2003)

23. Moens, M.F., Angheluta, R., Dumortier, J.: Generic technologies for single-and multi-document summarization. Inf. Process. Manag. **41**(3), 569–586 (2005)
24. Moens, M.F., Uyttendaele, C., Dumortier, J.: Abstracting of legal cases: the salomon experience. In: Proceedings of the 6th international conference on Artificial Intelligence and Law, pp. 114–122. ACM, Melbourne, Victoria, Australia (1997)
25. van Noord, G.: At last parsing is now operational. In: TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues Naturelles, pp. 20–42. Leuven, Belgium (2006)
26. van Noord, G., Schuurman, I., Bouma, G.: Lassy syntactische annotatie, revision 19053 (2010)
27. Paul, O., James, Y.: An introduction to duc-2004. In: Proceedings of the 4th Document Understanding Conference (DUC 2004), Boston, MA, USA (2004)
28. Radev, D., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. Comput. linguist. **28**(4), 399–408 (2002)
29. Shieber, S.: A uniform architecture for parsing and generation. In: Proceedings of the 12th COLING conference, Budapest (1988)
30. Teufel, S., Moens, M.: Summarizing scientific articles: experiments with relevance and rhetorical status. Comput. Linguist. **28**(4), 409–445 (2002)
31. Turner, J., Charniak, E.: Supervised and unsupervised learning for sentence compression. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 290–297. Association for Computational Linguistics, Ann Arbor, USA (2005)
32. Vandeghinste, V., Pan, Y.: Sentence compression for automated subtitling: a hybrid approach. In: Proceedings of the ACL Workshop on Text Summarization, Barcelona, Spain, pp. 89–95 (2004)
33. Vandeghinste, V., Tjong Kim Sang, E.: Using a parallel transcript/subtitle corpus for sentence compression. In: Proceedings of LREC 2004. Citeseer, Lisbon, Portugal (2004)
34. Velldal, E.: Empirical realization ranking. Ph.D. thesis, University of Oslo, Department of Informatics (2008)

Chapter 20
Generating, Refining and Using Sentiment Lexicons

Maarten de Rijke, Valentin Jijkoun, Fons Laan, Wouter Weerkamp, Paul Ackermans, and Gijs Geleijnse

20.1 Introduction

In this chapter, which is based on [7–9], we report on work on the generation, refinement and use of sentiment lexicons that was carried out within the DuOMAn project. The project was focused on the development of language technology to support online *media analysis*. In the area of media analysis, one of the key tasks is collecting detailed information about opinions and attitudes toward specific topics from various sources, both offline (traditional newspapers, archives) and online (news sites, blogs, forums). Specifically, media analysis concerns the following system task: given a topic and list of documents (discussing the topic), find all instances of attitudes toward the topic (e.g., positive/negative sentiments, or, if the topic is an organisation or person, support/criticism of this entity). For every such instance, one should identify the source of the sentiment, the polarity and, possibly, subtopics that this attitude relates to (e.g., specific targets of criticism or support). Subsequently, a (human) media analyst must be able to aggregate the extracted information by source, polarity or subtopics, allowing him to build support/criticism networks etc. [1]. Recent advances in language technology, especially in *sentiment analysis*, promise to (partially) automate this task.

Sentiment analysis is often considered in the context of the following two tasks:

M. de Rijke (✉) · F. Laan · W. Weerkamp
ISLA, University of Amsterdam, Amsterdam, Netherlands
e-mail: derijke@uva.nl; a.c.laan@uva.nl; w.weerkamp@uva.nl

V. Jijkoun
Textkernel BV, Amsterdam, Netherlands
e-mail: jijkoun@textkernel.nl

P. Ackermans · G. Geleijnse
Philips Research Europe, Eindhoven, Netherlands
e-mail: paul.ackermans@philips.com; gijs.geleijnse@philips.com

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_20,

- *Sentiment extraction*: given a set of textual documents, identify phrases, clauses, sentences or entire documents that express attitudes, and determine the polarity of these attitudes [11]; and
- *Sentiment retrieval*: given a topic (and possibly, a list of documents relevant to the topic), identify documents that express attitudes *toward this topic* [21].

How can technology developed for sentiment analysis be applied to media analysis? In order to use a *sentiment extraction* system for a media analysis problem, a system would have to be able to determine which of the extracted sentiments are relevant, i.e., it would not only have to identify targets of extracted sentiments, but also decide which targets are relevant for the topic at hand. This is a difficult task, as the relation between a *topic* (e.g., a movie) and specific targets of sentiments (e.g., acting or special effects in the movie) is not always straightforward, in the face of complex linguistic phenomena such as referential expressions ("... this beautifully shot *documentary*") or bridging anaphora ("the *director* did an excellent job").

In *sentiment retrieval*, on the other hand, the topic is initially present in the task definition, but it is left to the user to identify sources and targets of sentiments, as systems typically return a list of documents ranked by relevance and opinionatedness. To use a traditional sentiment retrieval system in media analysis, one would still have to manually go through ranked lists of documents returned by the system. To be able to support media analysis, we need to combine the specificity of (phrase- or word-level) sentiment analysis with the topicality provided by sentiment retrieval. Moreover, we should be able to identify sources and specific targets of opinions. Another issue is *evidence* for a system's decision. If the output of a system is to be used to inform actions, the system should present evidence, e.g., highlighting words or phrases that indicate a specific attitude. Most modern approaches to sentiment analysis, however, use various flavors of classification, where decisions (typically) come with confidence scores, but without explicit support.

In the first part of this chapter—Sects. 20.3–20.6—we focus on two of the problems identified above: (1) pinpointing evidence for a system's decisions about the presence of sentiment in text, and (2) identifying specific targets of sentiment. We address these problems by introducing a special type of lexical resource: a topic-specific subjectivity lexicon that indicates specific relevant targets for which sentiments may be expressed; for a given topic, such a lexicon consists of pairs (*syntactic clue*, *target*). We present a method for automatically generating a topic-specific lexicon for a given topic and query-biased set of documents. We evaluate the quality of the lexicon both manually and in the setting of an opinionated blog post retrieval task. We demonstrate that such a lexicon is highly *focused*, allowing one to effectively pinpoint evidence for sentiment, while being competitive with traditional subjectivity lexicons consisting of (a large number of) clue words.

In Sect. 20.7, we address the task of detecting on-topic subjectivity in text. Specifically, we want to (1) tell whether a textual document expresses an attitude (positive or negative) towards a specific topic, and moreover, (2) to find where exactly in the document it is expressed (up to a phrase or at least a sentence). The first task is in the area of *sentiment retrieval*. The simplest approach here consist

of two stages: first, we find texts that are on topic, then we filter out those without attitude [14]. A more elaborate approach is based on the assumption that documents are mixtures of two generative components, one "topical" and one "subjective" [17]. In practice, however, these components are not independent: a word that is neutral w.r.t. one topic can be a good subjectivity clue for another (e.g., compare *hard copy* and *hard problem*). Noticing this, Na et al. [20] generate a topic-specific list of possible clues, based on top relevant documents, and use this list for subjectivity filtering (reranking). In Sects. 20.3–20.6 we argue that such clues are specific not only to the topic, but to the exact target they refer to, e.g., when looking for opinions about a sportsman, *solid* is a good subjectivity clue in the phrase *solid performance* but not in *solid color*.

In Sect. 20.8 we explore the task of experience mining, where the goal is to gain insights into criteria that people formulate to judge or rate a product or its usage. We reveal several features that are likely to prove useful for automatic labeling via classification, over and above lexicon-based opinion spotting.

20.2 Related Work

Much work has been done in sentiment analysis. Here, we discuss work related to Sects. 20.3–20.6 of the chapter in four parts: sentiment analysis in general, domain- and target-specific sentiment analysis, product review mining and sentiment retrieval.

20.2.1 Sentiment Analysis

Sentiment analysis is often seen as two separate steps for determining subjectivity and polarity. Most approaches first try to identify subjective units (documents, sentences), and for each of these determine whether it is positive or negative. Kim and Hovy [11] select candidate sentiment sentences and use word-based sentiment classifiers to classify unseen words into a negative or positive class. First, the lexicon is constructed from WordNet: from several seed words, the structure of WordNet is used to expand this seed to a full lexicon. Next, this lexicon is used to measure the distance between unseen words and words in the positive and negative classes. Based on word sentiments, a decision is made at the sentence level. A similar approach is taken by Wilson et al. [30]: a classifier is learnt that distinguishes between polar and neutral sentences, based on a prior polarity lexicon and an annotated corpus. Among the features used are syntactic features. After this initial step, the sentiment sentences are classified as negative or positive; again, a prior polarity lexicon and syntactic features are used. The authors later explored the difference between prior and contextual polarity [31]: words that lose polarity in context, or whose polarity is reversed because of context. Riloff and Wiebe [24] describe

a bootstrapping method to learn subjective extraction patterns that match specific syntactic templates, using a high-precision sentence-level subjectivity classifier and a large unannotated corpus. In our method, we bootstrap from a subjectivity lexicon rather than a classifier, and perform a topic-specific analysis, learning indicators of subjectivity toward a specific topic.

20.2.2 Domain- and Target-Specific Sentiment

The way authors express their attitudes varies with the domain: An unpredictable movie can be positive, but unpredictable politicians are usually something negative. Since it is unrealistic to construct sentiment lexicons, or manually annotate text for learning, for every imaginable domain or topic, automatic methods have been developed. Godbole et al. [6] aim at measuring overall subjectivity or polarity towards a certain entity; they identify sentiments using domain-specific lexicons. The lexicons are generated from manually selected seeds for a broad domain such as *Health* or *Business*, following an approach similar to [11, 12]. All named entities in a sentence containing a clue from a lexicon are considered targets of sentiment for counting. Choi et al. [4] advocate a joint topic-sentiment analysis. They identify "sentiment topics," noun phrases assumed to be linked to a sentiment clue in the same expression. They address two tasks: identifying sentiment clues, and classifying sentences into positive, negative, or neutral. They start by selecting initial clues from SentiWordNet, based on sentences with known polarity. Next, the sentiment topics are identified, and based on these sentiment topics and the current list of clues, new potential clues are extracted. The clues can be used to classify sentences. Fahrni and Klenner [5] identify potential targets in a given domain, and create a target-specific polarity adjective lexicon. They find targets using Wikipedia, and associated adjectives. Next, the target-specific polarity of adjectives is determined using Hearst-like patterns. Kanayama and Nasukawa [10] introduce polar atoms: minimal human-understandable syntactic structures that specify polarity of clauses. The goal is to learn new domain-specific polar atoms, but these are not target-specific. They use manually-created syntactic patterns to identify atoms and coherency to determine polarity. In contrast to much of the work in the literature, we need to specialise subjectivity lexicons not for a domain and target, but for "topics."

20.2.3 Product Features and Opinions

Much work has been done on the task of mining product reviews, where the goal is to identify features of specific products (such as *picture*, *zoom*, *size*, *weight* for digital cameras) and opinions about these specific features in user reviews. Liu et al. [15] describe a system that identifies such features via rules learned from a manually

annotated corpus of reviews; opinions on features are extracted from the structure of reviews (which explicitly separate positive and negative opinions). Popescu and Etzioni [23] present a method that identifies product features for using corpus statistics, WordNet relations and morphological cues. Opinions about the features are extracted using a hand-crafted set of syntactic rules. Targets extracted in our method for a topic are similar to features extracted in review mining for products. Topics in our setting go beyond concrete products; the diversity and generality of possible topics makes it difficult to apply such supervised or thesaurus-based methods to identify opinion targets. Moreover, we directly use associations between targets and opinions to extract both.

20.2.4 Sentiment Retrieval

At TREC, the Text REtrieval Conference, there has been interest in a specific type of sentiment analysis: opinion retrieval. This interest materialised in 2006 [21], with the opinionated blog post retrieval task. Finding blog posts that are not just about a topic, but also contain an opinion on the topic, proves to be a difficult task [27,28]. Performance on the opinion-finding task is dominated by performance on the underlying document retrieval task (the topical baseline). Opinion finding is often approached as a two-stage problem: (1) identify documents relevant to the query, (2) identify opinions. In stage (2) one commonly uses either a binary classifier to distinguish between opinionated and non-opinionated documents or applies reranking of the initial result list using some opinion score. Opinion add-ons show only slight improvements over relevance-only baselines. The best performing opinion finding system at TREC 2008 is a two-stage approach using reranking in stage (2) [14]. The authors use SentiWordNet and a corpus-derived lexicon to construct an opinion score for each post in an initial ranking of blog posts. This score is combined with the relevance score, and posts are reranked according to this new score. We detail this approach in Sect. 20.6. Later, the authors use domain-specific opinion indicators [20], like "interesting story" (movie review), and "light" (notebook review). This domain-specific lexicon is constructed using feedback-style learning: retrieve an initial list of documents and use the top documents as training data to learn an opinion lexicon. Opinion scores per document are then computed as an average of opinion scores over all its words. Results show slight improvements (+3 %) on mean average precision.

20.3 Generating Topic-Specific Lexicons

In this section we describe how we generate a lexicon of subjectivity clues and targets for a given *topic* and a list of *relevant documents* (e.g., retrieved by a search engine for the topic). As an additional resource, we use a large background corpus

Table 20.1 Examples of subjective syntactic contexts of clue words (based on Stanford dependencies)

Clue word	Syntactic context	Target	Example
To like	Has direct object	*u2*	*I do still like U2 very much*
To like	Has clausal complement	*Criticize*	*I don't like to criticize our intelligence services*
To like	Has *about*-modifier	*Olympics*	*That's what I like about Winter Olympics*
Terrible	Is adjectival modifier of	*Idea*	*It's a terrible idea to recall judges for...*
Terrible	Has nominal subject	*Shirt*	*And Neil, that shirt is terrible!*
Terrible	Has clausal complement	*Can*	*It is terrible that a small group of extremists can ...*

of text documents of a similar style but with diverse subjects; we assume that the relevant documents are part of this corpus as well. As the background corpus, we used the set of documents from the assessment pools of TREC 2006–2008 opinion retrieval tasks (described in detail in Sect. 20.4). We use the Stanford lexicalised parser[1] to extract labeled dependency triples (*head, label, modifier*). In the extracted triples, all words indicate their category (*noun, adjective, verb, adverb,* etc.) and are normalised to lemmas. Figure 20.1 provides an overview of our method; below we describe it in more detail.

20.3.1 Step 1: Extracting Syntactic Contexts

We start with a general domain-independent prior polarity lexicon of 8,821 clue words [30]. First, we identify *syntactic contexts* in which specific clue words can be used to express attitude: we try to find how a clue word can be syntactically linked to targets of sentiments. We take a simple definition of the syntactic context: a single labeled directed dependency relation. For every clue word, we extract all syntactic contexts, i.e., all dependencies, in which the word is involved (as head or as modifier) in the background corpus, along with their endpoints. Table 20.1 shows examples of clue words and contexts that indicate sentiments. For every clue, we only select those contexts that exhibit a high entropy among the lemmas at the other endpoint of the dependencies.

Our entropy-driven selection of syntactic contexts of a clue word is based on the following assumption:

Assumption 1. In text, targets of sentiments are more diverse than sources of sentiments or other accompanying attributes such as location, time, manner, etc. Therefore targets exhibit higher entropy than other attributes.

[1] http://nlp.stanford.edu/software/lex-parser.shtml

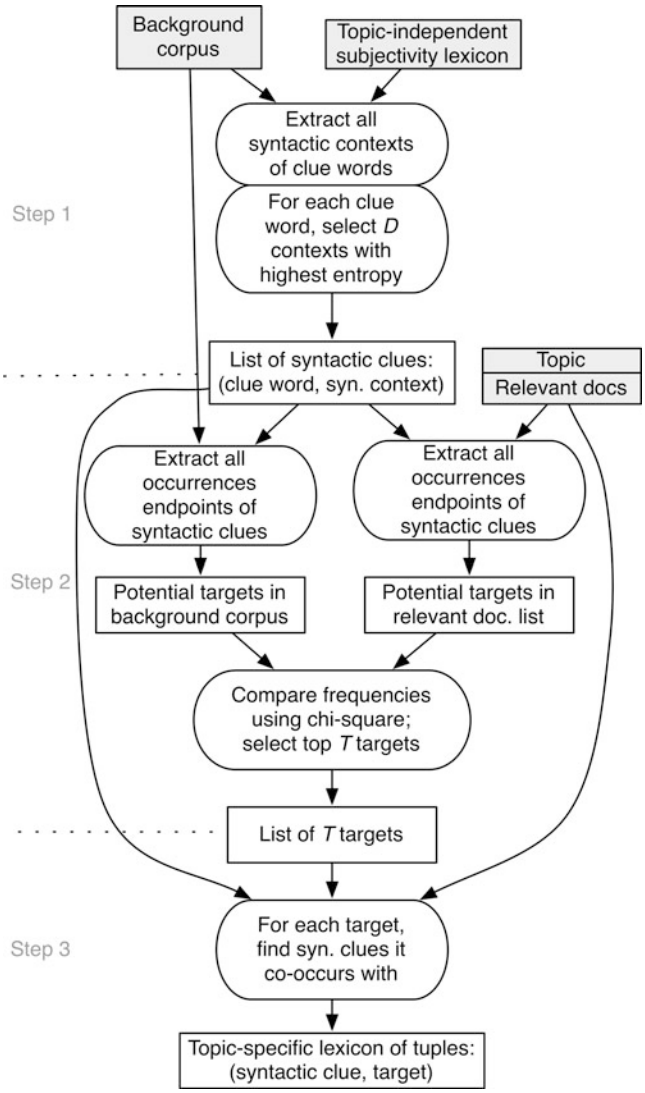


Fig. 20.1 Our method for learning a topic-dependent subjectivity lexicon

For every clue word, we select the top D syntactic contexts whose entropy is at least half of the maximum entropy for this clue. To summarise, at the end of Step 1 of our method, we have extracted a list of pairs (*clue word, syntactic context*) such that for occurrences of the clue word, the words at the endpoint of the syntactic dependency are likely to be targets of sentiments. We call such a pair a *syntactic clue*.

Table 20.2 Examples of targets extracted at Step 2

Topic "Relationship between Abramoff and Bush"
abramoff lobbyist scandal fundraiser bush fund-raiser republican prosecutor tribe swirl corrupt corruption norquist democrat lobbying investigation scanlon reid lawmaker dealings president
Topic "MacBook Pro"
macbook laptop powerbook connector mac processor notebook fw800 spec firewire imac pro machine apple powerbooks ibook ghz g4 ata binary keynote drive modem
Topic: "Super Bowl ads"
ad bowl commercial fridge caveman xl endorsement advertising spot advertiser game super essential celebrity payoff marketing publicity brand advertise watch viewer tv football venue

20.3.2 Step 2: Selecting Potential Targets

Here, we use the extracted syntactic clues to identify words that are likely to serve as specific targets for opinions about the topic in the relevant documents. In this work we only consider individual words as potential targets and leave exploring other options (e.g., NPs and VPs as targets) for future work. In extracting targets, we rely on the following assumption:

Assumption 2. The list of relevant documents contains a substantial number of documents on the topic which, moreover, contain sentiments about the topic.

We extract all endpoints of all occurrences of the syntactic clues in the relevant documents, as well as in the background corpus. To identify potential attitude targets in the relevant documents, we compare their frequency in the relevant documents to the frequency in the background corpus using the standard χ^2 statistics. This technique is based on the following assumption:

Assumption 3. Sentiment targets related to the topic occur more often in subjective context in the set of relevant documents, than in the background corpus. While the background corpus contains sentiments towards very diverse subjects, the relevant documents tend to express attitudes related to the topic.

For every potential target, we compute the χ^2-score and select the top T highest scoring targets. As the result of Steps 1 and 2, as candidate targets for a given topic, we only select words that occur in subjective contexts, and that do so more often than we would normally expect. Table 20.2 shows examples of extracted targets for three TREC topics (see below for a description of our experimental data).

20.3.3 Step 3: Generating Topic-Specific Lexicons

In the last step of the method, we combine clues and targets. For each target identified in Step 2, we take all syntactic clues extracted in Step 1 that co-occur with the target in the relevant documents. The resulting list of triples (*clue word, syntactic context, target*) constitute the lexicon. We conjecture that an occurrence of

a lexicon entry in a text indicates, with reasonable confidence, a subjective attitude towards the target.

20.4 Data and Experimental Setup

We consider two types of evaluation. In the next section, we examine the quality of the lexicons we generate. After that we evaluate lexicons quantitatively using the TREC Blog track benchmark. We apply our lexicon generation method to a collection of documents containing opinionated utterances: blog posts [16]. We perform two preprocessing steps [27, 28]: (1) when extracting plain text from HTML, we only keep block-level elements longer than 15 words (to remove boilerplate material), and (2) we remove non-English posts using TextCat[2] for language detection. We index the collection using Indri,[3] version 2.10 [13]. TREC 2006–2008 came with the task of *opinionated blog post retrieval* [21]. For each year a set of 50 topics was created, giving us 150 topics in total. Every topic comes with a set of relevance judgments: Given a topic, a blog post can be either (1) nonrelevant, (2) relevant, but not opinionated, or (3) relevant and opinionated. TREC topics consist of three fields (*title*, *description*, and *narrative*), of which we only use the *title* field: a query of 1–3 keywords.

We use standard evaluation measures for opinion retrieval: MAP (mean average precision), R-precision (precision within the top R retrieved documents, where R is the number of known relevant documents in the collection), MRR (mean reciprocal rank), P@10 and P@100 (precision within the top 10 and 100 retrieved documents). In the context of media analysis, recall-oriented measures such as MAP and R-precision are more meaningful than early precision-oriented measures. For the opinion retrieval task a document is considered relevant if it is on topic and contains opinions or sentiments towards the topic. We test for significant differences using a two-tailed paired t-test, and report on significant differences for $\alpha = 0.01$ ($^{\blacktriangle}$ and $^{\blacktriangledown}$), and $\alpha = 0.05$ ($^{\triangle}$ and $^{\triangledown}$). For the quantitative experiments in Sect. 20.6 we need a topical baseline: a set of blog posts potentially relevant to each topic. For this, we use the Indri retrieval engine, and apply the Markov Random Fields to model term dependencies in the query [19] to improve topical retrieval. We retrieve the top 1,000 posts for each query.

20.5 Qualitative Analysis of Lexicons

Lexicon size (the number of entries) and selectivity (how often entries match in text) of the generated lexicons vary depending on the parameters D and T introduced above. The two rightmost columns of Table 20.4 show the lexicon size

[2] http://odur.let.rug.nl/~vannoord/TextCat/

[3] http://www.lemurproject.org/indri/

Table 20.3 Posts with highlighted targets (*bold*) and subjectivity clues (*blue*) using topic-independent (*left*) and topic-specific (*right*) lexicons

There are some *tragic* moments like eggs freezing , and predators *snatching* the females and *little* ones-you know the whole *NATURE* thing . . . but this movie is *awesome*	There are some tragic moments l ike eggs freezing , and predators snatching the females and little ones-you know the whole NATURE thing . . . but this **movie** is *awesome*
Saturday was more errands, then spent the evening with Dad and Stepmum, and *finally* was *able* to see March of the Penguins, which was *wonderful*. Christmas Day was *lovely*, surrounded by family, *good* food and drink, and *little* L to play with	Saturday was more errands, then spent the evening with Dad and Stepmum, and finally was able to see March of the **Penguins**, which was *wonderful*. Christmas Day was lovely, surrounded by family, good food and drink, and little L to play with

and the average number of matches per topic. Because our topic-specific lexicons consist of triples (*clue word, syntactic context, target*), they actually contain more words than topic-independent lexicons of the same size, but topic-specific entries are more selective, which makes the lexicon more focused. Table 20.3 compares the application of topic-independent and topic-specific lexicons to on-topic blog text. We manually performed an explorative error analysis on a small number of documents, annotated using the smallest lexicon in Table 20.4 for the topic "March of the Penguins." We assigned 186 matches of lexicon entries in 30 documents into four classes: REL: sentiment towards a relevant target; CONTEXT: sentiment towards a target that is irrelevant to the topic due to context (e.g., opinion about a target "film", but referring to a film different from the topic); IRREL: sentiment towards irrelevant target (e.g., "game" for a topic about a movie); NOSENT: no sentiment at all. In total only 8 % of matches were manually classified as REL, with 62 % classified as NOSENT, 23 % as CONTEXT, and 6 % as IRREL. Among documents assessed as opinionated by TREC assessors, only 13 % did not contain matches of the lexicon entries, compared to 27 % of non-opinionated documents, which does indicate that our lexicon does attempt to separate non-opinionated documents from opinionated.

20.6 Quantitative Evaluation of Lexicons

In this section we assess the quality of the generated topic-specific lexicons numerically and extrinsically. To this end we deploy our lexicons to the task of opinionated blog post retrieval [21]. A commonly used approach to this task works in two stages: (1) identify topically relevant blog posts, and (2) classify these posts as being opinionated or not. In stage 2 the standard approach is to rerank the results from stage 1. We take this approach, as it has shown good performance in the past TREC editions [21] and is fairly straightforward to implement. Our experiments have two goals: to compare the use of topic-independent and topic-specific lexicons for the

Table 20.4 Evaluation of topic-specific lexicons applied to the opinion retrieval task, compared to the topic-independent lexicon. The two rightmost columns show the number of lexicon entries (avg. per topic) and the number of matches of lexicon entries in blog posts (avg. for top 1,000 posts)

Lexicon			MAP	R-prec	MRR	P@10	P@100	|Lexicon|	Hits per doc
No reranking			0.2966	0.3556	0.6750	0.4820	0.3666	–	–
Topic-independent			0.3182	0.3776	**0.7714**	**0.5607**	0.3980	8,221	36.17
D	T	S_{op}							
3	50	count	0.3191	0.3769	0.7276$^{\triangledown}$	0.5547	0.3963	2,327	5.02
3	100	count	0.3191	0.3777	0.7416	0.5573	0.3971	3,977	8.58
5	50	count	0.3178	0.3775	0.7246$^{\triangledown}$	0.5560	0.3931	2,784	5.73
5	100	count	0.3178	0.3784	0.7316$^{\triangledown}$	0.5513	0.3961	4,910	10.06
All	50	count	0.3167	0.3753	0.7264$^{\triangledown}$	0.5520	0.3957	4,505	9.34
All	100	count	0.3146	0.3761	0.7283$^{\triangledown}$	0.5347$^{\triangledown}$	0.3955	8,217	16.72
All	50	okapi	0.3129	0.3713	0.7247$^{\blacktriangledown}$	0.5333$^{\triangledown}$	0.3833$^{\triangledown}$	4,505	9.34
All	100	okapi	0.3189	0.3755	0.7162$^{\blacktriangledown}$	0.5473	0.3921	8,217	16.72
All	200	okapi	**0.3229**$^{\blacktriangle}$	**0.3803**	0.7389	0.5547	**0.3987**	14,581	29.14

opinionated post retrieval task, and to examine how settings for the parameters of the lexicon generation affect the empirical quality.

To rerank a list of posts retrieved for a given topic, we opt to use the method that showed best performance at TREC 2008. The approach taken by Lee et al. [14] linearly combines a (topical) relevance score with an opinion score for each post. For the opinion score, terms from a (topic-independent) lexicon are matched against the post content, and weighted with the probability of term's subjectivity. Finally, the sum is normalised using the Okapi BM25 framework. The final opinion score S_{op} is computed as in Eq. 20.1:

$$S_{op}(D) = \frac{Opinion(D) \cdot (k_1 + 1)}{Opinion(D) + k_1 \cdot (1 - b + \frac{b \cdot |D|}{avgdl})}, \tag{20.1}$$

where k_1, and b are Okapi parameters (set to their default values $k_1 = 2.0$, and $b = 0.75$), $|D|$ is the length of document D, and $avgdl$ is the average document length in the collection. The opinion score $Opinion(D)$ is calculated as $Opinion(D) = \sum_{w \in O} P(sub|w) \cdot n(w, D)$, where O is the set of terms in the sentiment lexicon, $P(sub|w)$ indicates the probability of term w being subjective, and $n(w, D)$ is the number of times term w occurs in document D. The opinion scoring can weigh lexicon terms differently, using $P(sub|w)$; it normalises scores to cancel out the effect of varying document sizes. We use the method described above, and plug in the MPQA polarity lexicon.[4] We compare the results of using this topic-independent lexicon to the topic-dependent lexicons our method generates,

[4] http://www.cs.pitt.edu/mpqa/

which are also plugged into the reranking of [14]. In addition to using Okapi BM25 for opinion scoring, we also consider a simpler method. As we observed in Sect. 20.5, our topic-specific lexicons are more selective than the topic-independent lexicon, and a simple number of lexicon matches can give a good indication of opinionatedness of a document: $S_{op}(D) = \min(n(O, D), 10)/10$, where $n(O, D)$ is the number of matches of the term of sentiment lexicon O in document D.

There are several parameters that we can vary when generating a topic-specific lexicon and when using it for reranking: D: the number of syntactic contexts per clue; T: the number of extracted targets; $S_{op}(D)$: the opinion scoring function; and α: the weight of the opinion score in the linear combination with the relevance score. Note that α does not affect the lexicon creation, but only how the lexicon is used in reranking. Since we want to assess the quality of lexicons, we factor out α by selecting the best setting for each lexicon (including the topic-independent) and each evaluation measure.

In Table 20.4 we present the results of evaluation of several lexicons in the context of opinionated blog post retrieval. First, we note that reranking using all lexicons significantly improves over the relevance-only baseline for all evaluation measures. When comparing topic-specific lexicons to the topic-independent one, most of the differences are not statistically significant, which is surprising given the fact that most topic-specific lexicons we evaluated are substantially smaller (see the two rightmost columns in the table). The smallest lexicon in Table 20.4 is seven times more selective than the general one, in terms of the number of lexicon matches per document. The only measure where the topic-independent lexicon consistently outperforms topic-specific ones, is MRR, which depends on a single relevant opinionated document high in a ranking. The general lexicon easily finds a "obviously subjective" posts (those with heavily used subjective words), but is not better at detecting less obvious ones, as indicated by the recall-oriented MAP and R-precision. Increasing the number of syntactic contexts considered for a clue word (parameter D) and the number of selected targets (parameter T) leads to substantially larger lexicons, but only gives marginal improvements when lexicons are used for opinion retrieval. This shows that our bootstrapping method is effective at filtering out non-relevant sentiment targets and syntactic clues. The choice of opinion scoring function (Okapi or raw counts) depends on the lexicon size: for smaller, more focused lexicons unnormalised counts are more effective; simple presence of a sentiment clue in text is a good indication of subjectivity. For larger lexicons an overall subjectivity scoring of texts has to be used, which can be hard to interpret for (media analysis) users.

20.7 Bootstrapping Subjectivity Detection

In Sects. 20.3–20.6 we have described a method for learning pairs (clue, target) for a given topic in an unsupervised manner, using syntactic dependencies between clues and targets. We go beyond the subjectivity lexicon generation methods from

Sects. 20.3–20.6, with the goal of improving subjectivity spotting. We directly evaluate the performance on the task of detecting on-topic subjectivity at the sentence level, not on sentiment retrieval with entire documents. Our method does not use a seed set.

20.7.1 Method

We start with a topic T (a textual description) and a set $R = \{d_1, \ldots, d_N\}$ of documents deemed relevant to T. The method uses a general-purpose list of subjectivity clues L (in our experiments, the well-known MPQA lexicon [29]). We use a background corpus BG of documents of a similar genre, covering many topics beside T. We use the Stanford syntactic parser to extract dependency relations in all sentences in all documents. Our method outputs a set of triples $\{(c_i, r_i, t_i)\}$, where c_i is a subjective clue, t_i a subjectivity target and r_i a dependency relation between the two words. We interpret an occurrence of such a triple as an indication of sentiment relevant to T, specifically directed at t_i.

We assume that a given topic can be associated with a number of related targets (e.g., opinions about a sportsman may cover such targets as *performance, reaction, serve,* etc.) and each target has a number of possible clues expressing attitude towards it (e.g., *solid performance*). We assume that clues and targets are typically syntactically related (e.g., the target *serve* can be a direct object of clue *to like*), and every clue has syntactic relations connecting it to possible targets (e.g., for *to like* only the direct object can be a target, but not the subject, a adverbial modifier, etc.).

20.7.1.1 Step 1: Initial Clue Scoring

For every possible clue $c \in L$ and every type of syntactic relation r that can originate from it in the background corpus, we compute a *clue score* $s_{clue}(c, r)$ as the entropy of words at the other endpoint of r in BG (normalised between 0 and 1 for all c and r). The clue score gives an initial estimate of how well (c, r) may work as a subjectivity clue. Here, we follow the intuition of Sects. 20.3–20.6: targets are more diverse than other syntactic neighbors of clues.

20.7.1.2 Step 2: Target Scoring

For every word $t \in R$ we determine its target score that tells us how likely t is an opinion target related to topic T. Targets are words that occur unusually often in subjective contexts in relevant documents. First, we compute $C_R(t) = \sum s_{clue}(c, r)$ for all occurrences of the syntactic relation r between words c and t in corpus R. Similarly, we compute $C_{BG}(t)$ for the background corpus BG. We view $C_R(\cdot)$ and $C_{BG}(\cdot)$ as (weighted) counts, and compute a parsimonious language model $p_R(\cdot)$

Table 20.5 Test results on a sentence classification task

Method				P	R	F_1
Method of Sects. 20.3–20.6				0.23	0.31	0.26
R	K	N	M			
$r + 100$	4	10	50	0.42	0.13	0.20
$r + 100$	4	20	50	0.45	0.17	0.25
$r + 100$	4	30	50	0.35	0.26	0.28
$r + 100$	4	40	50	0.32	0.29	0.30
$r + 100$	4	50	50	0.20	0.30	0.24
$r + 100$	4	60	50	0.19	0.32	0.24
$r + 100$	4	70	50	0.14	0.35	0.20
$r + 100$	4	40	30	0.32	0.21	0.25
$r + 100$	4	40	40	0.32	0.23	0.27
$r + 100$	4	40	50	0.32	0.29	0.30
$r + 100$	4	40	60	0.30	0.29	0.29
$r + 100$	4	40	70	0.29	0.30	0.29
$r + 100$	4	40	50	0.32	0.29	0.30
100	4	40	50	0.27	0.22	0.24
r	4	40	50	0.21	0.17	0.19

using a simple EM algorithm [18]. We also compute a language model $p_B G(\cdot)$ from counts $C_{BG}(\cdot)$ by simple normalisation. Finally, we define the target score of a word t as the likelihood that the occurrence of t in R comes from $p_R(\cdot)$ rather than $p_B G(\cdot)$:

$$s_{tgt}(t) = \frac{\gamma \cdot p_{tgt}(t)}{\gamma \cdot p_{tgt}(t) + (1 - \gamma) \cdot p_{BG}(t)}.$$

20.7.1.3 Step 3: Clue Scoring

Mirroring Step 2, we now use target scores to compute better estimates for clue scores. Here, our intuition is that good subjectivity clues are those that occur unusually often near possible opinion targets for a given topic. The computation is similar to Step 2, with $s_{clue}(c, r)$ and $s_tgt(t)$ interchanged: we compute weighted counts, a parsimonious model and, finally, the updated $s_{clue}(c, r)$. Now, we iterate Step 2 and Step 3, each time updating $s_{tgt}(\cdot)$ and $s_{clue}(\cdot, \cdot)$, respectively, based on the values at the previous iteration. After K iterations we select N targets and M pairs (clue, relation) with the highest scores. We check which of the N targets co-occur with which of the M clues in R.

20.7.2 Experiments and Results

We evaluate different versions of our method on the following sentence classification task: for a given topic and a list of documents relevant to the topic, we

need to identify sentences that express opinions relevant to the topic. We compute precision, recall and F-score for detection of relevant opinionated sentences. We use the NTCIR-6 [25] and NTCIR-7 [26] Opinion Analysis datasets, containing judgements for 45 queries and 12,000 sentences. In order to understand how the quality of relevant documents affects the performance of the method, we selected R to be (1) R_{100}: top 100 document retrieved from the NTCIR-6/7 English collection using Lucene, (2) R_r: only documents with at least one relevant (not necessarily opinionated) sentence as identified by NTCIR annotators, and (3) R_{r+100} the union of (1) and (3). We ran the method with different numbers of iterations (K), selected targets (N) Table 20.5 shows the results, the overall performance stabilises at $K \leq 5$. The table included above shows the evaluation results. We see that reducing the number of selected targets (N) improves precision but harms recall. Changing the number of selected clues (M) has little effect on precision: since for detecting opinionatedness we combine clues with targets, noise in clues does not necessarily lead to drop in precision. Overall, we notice that in the best setting ($K = 4$, $N = 40$, $M = 50$) the method outperforms the method described in Sects. 20.3–20.6 (significantly, at $p = 0.05$, using t-test). Performance of the method varies substantially per topic (F_1 between 0.13 and 0.48), but the optimal values for parameters are stable for high-performing topics (with $F_1 > 0.26$).

20.8 Mining User Experiences from Online Forums

We change tack again and report on an exploratory study. It touches on an important step after the initial groundwork laid down by lexicon generation and refinement of the type described so far: mining user experiences. Let us provide some background. Recent years have shown a large increase in the usage of content creation platforms aimed at the general public. User generated data contains emotional, opinionated, sentimental, and personal posts. This feature makes it an interesting data source for exploring new types of text analysis, as is shown by research on sentiment analysis [22], opinion retrieval [21], and mood detection [2]. We introduce the task of *experience mining*. Here, the goal is to gain insights into criteria that people formulate to judge or rate a product or its usage. We focus on reports of experiences with products.

20.8.1 Motivation

Our main use-case is user-centered design for product development. User-centered design [3] is an innovation paradigm where users of a product are involved in each step of the research and development process. The first stage of the product design process is to identify unmet needs and demands of users for a specific product or a class of products. Although statements found in such platforms may not always be representative for the general user group, they can accelerate user-centered design.

20.8.2 Experience Mining

Experiences are particular instances of personally encountering or undergoing something. We want to identify experiences about a specific *target product*, that are *personal*, involve an *activity* related to the target and, moreover, are accompanied by *judgements or evaluative statements*. Experience mining is related to sentiment analysis and opinion retrieval, in that it involves identifying attitudes; the key difference is, however, that we are looking for *attitudes towards specific experiences* with products, not attitudes towards the products themselves.

20.8.3 An Explorative Study

To assess the feasibility of automatic experience mining, we carried out an explorative study: we asked human assessors to find experiences in actual forum data and then examined linguistic features likely to be useful for identifying experiences automatically. We acquired data by crawling two forums on shaving,[5] with 111,268 posts written by 2,880 users. Two assessors searched for posts on five specific target products using a standard keyword search, and labeled each result post as: (1) reporting no experience, or (2) reporting an off-target experience, or (3) reporting an on-target experience. Posts should be marked as reporting an experience only if (1) the author explicitly reports his or someone else's (a concrete person's) use of a product; and (2) the author makes some conclusions/judgements about the experience. In total, 203 posts were labeled, with 101 posts marked as reporting an experience by at least one assessor (71 % of those an on-target experience). The inter-annotator agreement was 0.84, with Cohen's $\kappa = 0.71$. If we merge on- and off-target experience labels, the agreement is 0.88, with $\kappa = 0.76$. The high level of agreement demonstrates the validity of the task definition. We considered a number of linguistic features and compared posts reporting experience (on- or off-target) to the posts with no experience. Table 20.6 lists the features and the comparison results. The subjectivity score is lower for experience posts: our task is indeed different from sentiment retrieval! Experience posts are on average twice as long as non-experience posts and contain more sentences with pronoun *I*. They also contain more content (non-modal) verbs, especially past tense verbs. Table 20.7 presents an analysis of the verb use. Experience posts contain more verbs referring to concrete actions than to attitude and perception. It remains to be seen whether this observation can be quantified using resources such as standard semantic verb classification (*state, process, action*), WordNet verb hierarchy or FrameNet semantic frames.

[5] http://www.shavemyface.com and http://www.menessentials.com/community.

Table 20.6 Comparison of surface features; $p(\cdot)$ denotes probability

	Mean and deviation in posts with/without experience	
Feature	*With*	*Without*
Subjectivity score[6]	0.07 ± 0.23	0.17 ± 0.35
Polarity score[6]	0.87 ± 0.30	0.77 ± 0.38
#words per post	102.57 ± 80.09	52.46 ± 53.24
#sentences per post	6.00 ± 4.16	3.34 ± 2.33
# words per sentence	17.07 ± 4.69	15.71 ± 7.61
#questions per post	0.32 ± 0.63	0.54 ± 0.89
p(post contains question)	0.25 ± 0.43	0.33 ± 0.47
#*I*'s per post	5.76 ± 4.75	2.09 ± 2.88
#*I*'s per sentence	1.01 ± 0.48	0.54 ± 0.60
p(sentence in post contains *I*)	0.67 ± 0.23	0.40 ± 0.35
#non-modal verbs per post	19.62 ± 15.08	9.82 ± 9.57
#non-modal verbs per sent.	3.30 ± 1.18	2.82 ± 1.37
#modal verbs per sent.	0.22 ± 0.22	0.26 ± 0.36
Fraction of past-tense verbs	0.26 ± 0.17	0.17 ± 0.19
Fraction of present tense verbs	0.42 ± 0.18	0.41 ± 0.23

Table 20.7 Frequent past tense verbs following *I* with relative frequencies

In posts with experience	In posts without experience
Used 0.15, found 0.09, bought 0.07, tried 0.07, got 0.07, went 0.07, started 0.05, switched 0.04, liked 0.03, decided 0.03	Got 0.09, thought 0.09, switched 0.06, meant 0.06, used 0.06, went 0.06, ignored 0.03, quoted 0.03, discovered 0.03, heard 0.03

20.9 Conclusion

We started this chapter by describing a bootstrapping method for deriving a topic-specific lexicon from a general purpose polarity lexicon. We evaluated the quality of the lexicons generated by our method both manually and using a TREC Blog track test set for opinionated blog post retrieval. Although the generated lexicons can be an order of magnitude more selective, they maintain, or even improve, the performance of an opinion retrieval system. In future work, we want to look at more complex syntactic Choi et al. [4] report that many errors are due to exclusive use of unigrams. We also want to extend potential opinion targets to include multi-word phrases (NPs and VPs).

Second, in this chapter we also described a method for automatically generating subjectivity clues for a specific topic and a set of (relevant) document, evaluating it on the task of classification of sentences w.r.t. subjectivity, demonstrating improvements over previous work. Here, we plan to incorporate more complex syntactic

[6]Computed using LingPipe: http://alias-i.com/lingpipe.

patterns in our clues (going beyond word-word relations) and study the effect of user feedback with the view of implementing an interactive system.

Finally, we explored the novel task of experience mining. Users of products share their experiences, and mining these could help define requirements for next-generation products. We developed annotation guidelines for labeling experiences, and used them to annotate data from online forums. An initial exploration revealed multiple features that might prove useful for automatic labeling via classification.

Acknowledgements In addition to funding by the STEVIN programme, this research was also partially supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 380-70-011, 727.011.005, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP project funded by the CLARIN-NL program, under COMMIT project Infiniti and by the ESF Research Network Program ELIAS.

References

1. Altheide, D.: Qualitative Media Analysis. Sage, Thousand Oaks (1996)
2. Balog, K., Mishne, G., de Rijke, M.: Why are they excited?: identifying and explaining spikes in blog mood levels. In: EACL '06, Trento, pp. 207–210 (2006)
3. Buxton, B.: Sketching User Experiences: Getting the Design Right and the Right Design. Morgan Kaufmann, San Francisco (2007)
4. Choi, Y., Kim, Y., Myaeng, S.-H.: Domain-specific sentiment analysis using contextual feature generation. In: TSA '09, pp. 37–44. ACM, New York (2009)
5. Fahrni, A., Klenner, M.: Old wine or warm beer: target-specific sentiment analysis of adjectives. In: AISB 2008 Convention, Aberdeen, pp. 60–63 (2008)
6. Godbole, N., Srinivasaiah, M., Skiena, S.: Large-scale sentiment analysis for news and blogs. In: ICWSM '07, Denver (2007)
7. Jijkoun, V., de Rijke, M.: Bootstrapping subjectivity detection. In: SIGIR '11, Beijing (2011)
8. Jijkoun, V., de Rijke, M., Weerkamp, W.: Generating focused topic-specific sentiment lexicons. In: ACL '10, Uppsala (2010a)
9. Jijkoun, V., de Rijke, M., Weerkamp, W., Ackermans, P., Geleijnse, G.: Mining user experiences from online forums: an exploration. In: NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, Los Angeles (2010b)
10. Kanayama, H., Nasukawa, T.: Fully automatic lexicon expansion for domain-oriented sentiment analysis. In: EMNLP '06, Sydney, pp. 355–363 (2006)
11. Kim, S., Hovy, E.: Determining the sentiment of opinions. In: COLING 2004, Geneva (2004)
12. Kim, Y., Choi, Y., Myaeng, S.-H.: Generating domain-specific clues using news corpus for sentiment classification. In: ICWSM '10, Washington, DC (2010)
13. Lavrenko, V., Croft, B.: Relevance-based language models. In: SIGIR '01, New Orleans (2001)

14. Lee, Y., Na, S.-H., Kim, J., Nam, S.-H., Jung, H.-Y., Lee, J.-H.: KLE at TREC 2008 blog track: blog post and feed retrieval. In: TREC 2008, Gaithersburg (2008)
15. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: WWW '05, Chiba (2005)
16. Macdonald, C., Ounis, I.: The TREC Blogs06 collection: creating and analysing a blog test collection. Technical Report TR-2006-224, University of Glasgow (2005)
17. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: modeling facets and opinions in weblogs. In: WWW '07, Banff, pp. 171–180 (2007)
18. Meij, E., Weerkamp, W., Balog, K., de Rijke, M.: Parsimonious relevance models. In: SIGIR '08, Singapore (2008)
19. Metzler, D., Croft, W.B.: A markov random feld model for term dependencies. In: SIGIR '05, Salvador, pp. 472–479 (2005)
20. Na, S.-H., Lee, Y., Nam, S.-H., Lee, J.-H.: Improving opinion retrieval based on query-specific sentiment lexicon. In: ECIR '09, Toulouse, pp. 734–738 (2009)
21. Ounis, I., Macdonald, C., de Rijke, M., Mishne, G., Soboroff, I.: Overview of the TREC 2006 blog track. In: TREC 2006, Gaithersburg (2007)
22. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**, 1–135 (2008)
23. Popescu, A.-M., Etzioni, O.: Extracting product features and opinions from reviews. In: HLT/EMNLP '05, Vancouver (2005)
24. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: EMNLP '03, Sapporo, Japan (2003)
25. Seki, Y., Evans, D.K., Ku, L.-W., Chen, H.-H., Kando, N., Lin, C.-Y.: Overview of opinion analysis pilot task at NTCIR-6. In: NTCIR-6, Tokyo (2007)
26. Seki, Y., Evans, D.K., Ku, L.-W., Sun, L., Chen, H.-H., Kando, N.: Overview of multilingual opinion analysis task at NTCIR-7. In: NTCIR-7, Tokyo (2008)
27. Weerkamp, W., Balog, K., de Rijke, M.: A generative blog post retrieval model that uses query expansion based on external collections. In: ACL-IJCNLP 2009, Singapore (2009)
28. Weerkamp, W., de Rijke, M.: Credibility improves topical blog post retrieval. In: ACL-08: HLT, Columbus, pp. 923–931 (2008)
29. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Lang. Resour. Eval. **39**, 165–210 (2005)
30. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT '05, Vancouver, Canada, pp. 347–354 (2005)
31. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. Comput. Linguist. **35**(3), 399–433 (2009)

Part V
And Now

Chapter 21
The Dutch-Flemish HLT Agency: Managing the Lifecycle of STEVIN's Language Resources

Remco van Veenendaal, Laura van Eerten, Catia Cucchiarini, and Peter Spyns

21.1 Introduction

The development and availability of human language technologies is considered crucial for a language to be able to survive in the information society. Since Dutch is a so-called mid-sized language [11, 12] with a relatively small market, companies tend to be reluctant to invest in the development of resources for such a language. If they do, the resulting data are not always made available to researchers or other companies at affordable prices. Hence governmental support is required.

The Dutch Language Union (NTU – Nederlandse Taalunie[1]), an intergovernmental organisation established by Belgium and the Netherlands, has stated as one of its priorities the promotion of the development of Human Language Technology (HLT) for the Dutch language [3, 6]. In co-operation with the relevant ministries and organisations in Belgium and the Netherlands, the NTU set up a number of initiatives (cf. Chap. 2, page 21). Two of these are particularly significant: the STEVIN research programme [13] and the Dutch-Flemish Human Language Technology Agency (HLT Agency, TST-Centrale in Dutch) [2]. In addition, these two initiatives are related as it was stipulated that the language resources developed within the STEVIN programe would later be handed over to the HLT Agency for subsequent management, maintenance and distribution.

[1] See http://taalunieversum.org/taalunie/how_can_we_help_you_/index.php.

R. van Veenendaal (✉) · L. van Eerten
Dutch-Flemish HLT Agency (TST-Centrale), Institute for Dutch Lexicology, Matthias de Vrieshof 2-3, 2311 BZ Leiden, The Netherlands
e-mail: remco.vanveenendaal@inl.nl; laura.vaneerten@inl.nl

C. Cucchiarini · P. Spyns
The Nederlandse Taalunie, Lange Voorhout 19, 2514 EB Den Haag, The Netherlands
e-mail: ccucchiarini@taalunie.org; pspyns@taalunie.org

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_21, © The Author(s) 2013

The establishment of a central repository for managing LRs prevents LRs developed with public money from becoming obsolete and therefore useless. Resources that are not maintained quickly lose value. In the past, official bodies such as ministries and research organisations used to only finance the development of LRs and did not have a clear policy on what should happen to these materials once the projects had been completed. Universities rarely have the resources to do maintenance work on completed projects, and knowledge is sometimes lost when experts switch jobs. It was against this background that the idea of having one central repository for digital language resources in the Dutch language area was conceived.

Having one organisation that is responsible for managing LR lifecycles creates higher visibility and better accessibility. Having a local one-stop-shop for Dutch LRs leads to more re-use of these LRs. Synergistic use of manpower and means is efficient and cost-reducing, compared to, for example, having several (smaller) organisations that only take care of certain aspects of LR lifecycles instead of their complete lifecycles. Furthermore, combining resources and bringing together different kinds of expertise creates surplus value. This can result in improved versions of datasets and new insights into potential use(r)s of LRs.

In this chapter, the HLT Agency is presented mainly from the perspective of its contribution to the STEVIN programme. In Sect. 21.2, the organisational set-up of the HLT Agency is described. Section 21.3 explains how the HLT Agency manages the lifecycle of STEVIN results. In Sect. 21.4, target groups and users are presented. Section 21.5, discusses new challenges and possible directions for the HLT Agency after the completion of the STEVIN programme. Section 21.6 concludes the chapter and points to future perspectives.

21.2 The Flemish-Dutch HLT Agency

The HLT Agency is an initiative of the NTU, from which it currently receives an annual subsidy of 450,000 euros. Additional financing comes mostly from projects, support and licence fees. The HLT Agency can be considered a non-profit, government-funded initiative. It is currently hosted at the Institute for Dutch Lexicology (Instituut voor Nederlandse Lexicologie, INL), a Dutch-Flemish organisation with offices in the Netherlands (Leiden) and Belgium (Antwerp).

The mission given by the NTU to the HLT Agency was to maintain, manage and distribute LRs for Dutch, in particular those owned by the NTU, which include the STEVIN results. This also implies clearing IPR issues with the suppliers of the LR at the in-take (acquisition licences – cf. Sect. 21.3.1) and safe-guarding the interests of the owners of the LR towards users of the resources (distribution licences – cf. Sect. 21.3.4). STEVIN is not the only source of LRs that are hosted by the HLT Agency. Aside from STEVIN, the HLT Agency tries to collect high-quality digital Dutch LRs from funding institutions such as the NTU, the INL, funding agencies like the NWO and third parties such as (individual) researchers,

universities and some other organisations and foundations. In fact, the first LRs the HLT Agency acquired came from previous government-funded projects (the Spoken Dutch Corpus [10], the Referentiebestand Nederlands,[2] both now property of the NTU, and several lexical resources of the INL).

The HLT Agency's core team consists of a project manager, three linguists, and a co-worker who takes care of all distribution procedures. Each linguist is responsible for a particular group of LRs. Additionally, one of the linguists doubles as a communications officer, another as manager of the licences. Most employees work part-time. In addition to the core team, one programmer, two computational linguists and one system administrator contribute to the HLT Agency part-time. They are regular employees of the INL and support the HLT Agency with technical work on LRs, such as updates or tailor-made versions of LRs. .

Other repositories for managing LRs exist (cf. [8, p. 39] for an overview). Two examples are DANS (in the Netherlands) and ELDA (in Europe):

- Data Archiving and Networked Services (DANS),[3] an institute of the Royal Netherlands Academy of Arts and Sciences (KNAW)[4] and the Netherlands Organisation for Scientific Research (NWO),[5] promotes sustained access to digital research data. For this purpose, DANS encourages researchers to archive and reuse data in a sustained manner, e.g., through an online archiving system.
- The European Language resources Distribution Agency (ELDA)[6] is the operational body of the European Language Resources Association (ELRA)[7] and was set up to identify, classify, collect, validate and produce language resources. ELDA is also involved in HLT evaluation campaigns.

Whereas DANS focuses on all research areas and ELDA on LRs in many languages, it is the HLT Agency's specific mission to take care of Dutch digital LRs in order to strengthen the position of the Dutch language in the information society. The HLT Agency therefore focuses on digital Dutch LRs (as opposed to ELDA) and needs to ensure that LRs are not only made available (as opposed to DANS), but also kept up-to-date and usable. This means the HTL Agency takes care of the management of the entire lifecycle of LRs, including maintenance and support. Creating new resources and performing evaluation campaigns currently do not belong to the mission of the HTL Agency (as opposed to ELDA).

[2] See http://www.tst-centrale.org/images/stories/producten/documentatie/rbn_documentatie_nl.pdf

[3] See http://www.dans.knaw.nl

[4] See http://www.knaw.nl

[5] See http://www.nwo.nl

[6] See http://www.elda.org

[7] See http://www.elra.info

21.3 Managing the Lifecycle of STEVIN Results

The HLT Agency distinguishes five different phases in the lifecycle management process: acquisition, management, maintenance, distribution and support. These phases are described in the following subsections.

21.3.1 Acquisition

From the very start of the STEVIN programme, the funding partners agreed that the NTU would become the owner of the LRs developed within the STEVIN projects and that these would be transferred to the HLT Agency. The rationale behind this decision was to ensure optimal LR accessibility (see below). STEVIN project proposals included a description of the potential future use of the resulting resources and of the contribution to the overall STEVIN aims (cf. Chap. 2, page 25). A positive review and subsequent funding of the project implied that the resources were worthwhile to be maintained and subsequently distributed. Therefore the process of acquisition was to a large extent fixed. The HLT Agency focused on settling issues concerning intellectual property rights (cf. Sect. 21.3.1.1) and checking the quality of the LRs delivered by the STEVIN projects (cf. Sect. 21.3.1.2).

21.3.1.1 Intellectual Property Rights

In general, the intellectual property rights (IPR) of the LRs developed and completed in the STEVIN programme (the foreground knowledge) were transferred to the NTU. However, if pre-existing software tools or data (background knowledge) were used in the projects, the rights on the background knowledge would remain with the original IPR-holder. If existing open source was re-used and improved, the resulting STEVIN LR would also become open source (as was the case for e.g., the STEVINcanPRAAT project – cf. Chap. 5, page 79).

Transferring all rights to one organisation, c.q. the NTU, has several practical advantages. IPR issues such as setting up licences and price negotiations can be dealt with more efficiently, both in terms of money and time; the more IPR holders, the more rights and responsibility issues. It is also a guarantee that IPR issues are dealt with in a legally sound manner, which in turn leads to considerably fewer restrictions on LR availability.

With regard to the data themselves, it was primarily the responsibility of the STEVIN projects to settle IPR issues such as copyright on texts and usage of speech data. The project proposals had to include a section on how IPR issues would be taken care of. The role of the HLT Agency was to provide assistance to projects during the process. As a result, the HLT Agency was involved in the process of settling IPR issues from an early stage.

Altogether, corpus creation projects like SoNaR (cf. Chap. 13, page 219) and DPC (cf. Chap. 11, page 185) have resulted in more than 150 signed data acquisition licences. Tailor-made versions of these licences were made available after discussing requests from data providers with the project team, a lawyer and the NTU. One example of a tailor-made version is a licence for publishers who are willing to provide data for research purposes only (while the standard licence also includes the use of the data for commercial purposes). The HLT Agency [8] acted as signing party for acquisition licences on behalf of the NTU.

21.3.1.2 Evaluation and Validation

STEVIN requested projects to have their deliverables externally evaluated. Evaluation is necessary to gauge the LR's quality and potential for re-use.

After completion of the projects, the evaluation reports were handed over to the HLT Agency, together with the LRs. The HLT Agency monitors the value indicators for the purpose of prioritising future maintenance work. In the case of e.g. the IRME (cf. Chap. 12, page 201), JASMIN-CGN (cf. Chap. 3, page 43) and Cornetto (cf. Chap. 10, page 165) projects, conducting an early external evaluation resulted in the delivery of improved project results at the end of the project.

The HLT Agency's active contribution to the validation process is limited to technical checks. All data are validated (e.g., against XML schemas) and the quality and completeness of the documentation is thoroughly checked. In the case of software the binaries are tested on the supported platforms, using the accompanying documentation to create test cases. Source codes are compiled to (object codes and linked into) testable binaries. If the data validation produces significant errors, software does not execute or does not work as expected (according to the documentation), it is primarily the LR provider's task to fix any problems and resubmit the fixed LR to the HLT Agency.[9]

21.3.2 Management

The LRs delivered by STEVIN projects are stored and backed up on servers hosted and maintained by the INL, the HLT Agency's hosting institute. Where needed, LRs are stored in version control software like Subversion.[10] Our archive and "production line system" [4] servers currently contain 1.5 terabyte and over 60 LRs.

---

[8] In fact, the INL signs the licences as the INL is the HLT Agency's legal entity.

[9] This practice is in line with the Data Seal of Approval guidelines (see http://www.datasealofapproval.org), adopted by the HLT Agency.

[10] See http://subversion.apache.org/. One reason for setting up Subversion for an LR is collaborative maintenance work.

21.3.3 Maintenance

The STEVIN LRs, including all accompanying deliverables (project proposals, reports and documentation), are stored and backed up by the HLT Agency in their original form. A separate distribution version is prepared, consisting for example of the LR and relevant user documentation. If the LR provider agrees, the evaluation report is included. Periodically, the HLT Agency checks if LRs need maintenance, e.g. for the purpose of standardisation or when they risk disuse due to incompatibility with new operating systems. Also actual use and peer reviews (user feedback) of the LRs give indications of whether or not LRs need maintenance. The HLT Agency distinguishes between minor and major maintenance.

Minor Maintenance The goal of minor maintenance is to keep resources usable, which means fixing critical bugs, updating manuals and documentation, upgrading formats to newer versions of the standard(s) used, etc. Minor maintenance is done by the HLT Agency itself. Periodically, the HLT Agency checks if LRs require minor maintenance and starts the work after having consulted the owner/supplier of the LR. Feedback from users is included in these maintenance checks. The result of minor maintenance is usually a patch or update of an LR. News on any updated versions is published, so that users can request an update for free.

Major Maintenance Major maintenance consists in significantly improving or expanding a resource. Therefore, major maintenance usually requires additional funding and cooperation with the developers and external experts. Information and advice on which LRs should be improved or expanded can be gathered from the various advisory committees that assist the NTU and the HTL Agency and from user feedback collected by the service desk. Major maintenance work usually results in a new version of an LR, rather than a patch or update. News on any new versions is published, for which all users must accept a (new) licence.

Minor and Major Maintenance in Practice Below we present some examples of maintenance on to the STEVIN IRME (cf. Chap. 12, page 201) and Cornetto (cf. Chap. 10, page 165) projects:

- IRME
  - In the final stage, but before the end of the project, version 1.0 of the DuELME resource (Dutch Electronic Lexicon of Multiword Expressions) was updated by the project members at Utrecht University after they had received the external validation report. The resulting version 1.1 was made available to the HLT Agency for further distribution.
  - The HLT Agency improved the Web interface for the lexicon by making minor adjustments in functionality and display to the search tool and by optimising the MacOSX-based DuELME web interface for Windows (minor maintenance).

  - Utrecht University and the HLT Agency converted the DuELME lexicon to LMF (Lexical Markup Framework) within the CLARIN-NL[11] project DuELME-LMF. This became version 2.0 of the resource (major maintenance). This version was also made available in the CLARIN infrastructure.

- Cornetto
  - In response to user requests, intermediate versions of the Cornetto database (Combinatorial and Relational Network as Tool kit for Dutch Language Technology; a lexical semantic database for Dutch) were made available during the project. The HLT Agency took care of the licences and the project team, led by the Free University Amsterdam (VUA), distributed and supported the database.
  - VUA and the HLT Agency improved version 1.0 of the Cornetto database to versions 1.2, 1.21 and 1.3 (minor maintenance).
  - Currently, VUA is significantly improving the Cornetto database. For example, sentiment values and text corpus references are being added for each word meaning. This work will result in version 2.0, to be released by the HLT Agency early 2012.
  - VUA and the HLT Agency will work on a further improved version of the Cornetto database in the CLARIN-NL Cornetto-LMF-RDF project (major maintenance). As a result, Cornetto will also become available within the CLARIN infrastructure, in LMF, RDF and SKOS formats.

21.3.4 Distribution

The HLT Agency makes the LRs available for users through a web shop[12]: users order an LR from the web shop and receive the LR after accepting an end user agreement. Most of the LRs are available as a downloadable file or through a web interface. Larger LRs are distributed off-line on DVDs or a hard disk. In the case of off-line distribution, the HLT Agency charges a small handling and shipping fee.[13]

The terms and conditions for the use of LRs made available by the HLT Agency are defined in (distribution) licence agreements. These agreements were written with the specific goals of the HLT Agency in mind: they have to stimulate the reuse of the LRs, but also support the idea of a central location where LRs are made and kept available. Feedback from users, stakeholders and legal experts has helped us improve and standardise the licences over the years. Other distribution centres, with different goals, apply other terms and conditions – e.g., [5].

[11] See http://www.clarin.nl

[12] A new HLT Agency web site with web shop has been launched in September 2011.

[13] Currently 50 euros are charged for (a set of one or more) DVDs and 100 euros for a hard disk.

In order to strengthen the position of the Dutch language in today's information society, it is necessary to stimulate the use of Dutch in research, education and commercial end user applications. This is considered more important than financial return on investment. It implies that the HLT Agency, also due to the relatively limited size of the Dutch language area, is not supposed to become self-sustainable (as opposed to e.g., ELDA) – cf. Sect. 21.2. In short, three licensing schemes are available:

- Single licensing (non-commercial or open-source licence only)
- Dual licensing (non-commercial and commercial licence)
- Dual licensing (open-source or commercial licence)

Open-source licences have to be commonly-accepted and non-viral open-source licences. Various possibilities and variants exist [9]. Our non-commercial licences are for non-commercial use by non-commercial organisations. There is no licence fee attached to non-commercial licences (apart from incidental exceptions due to third-party rights). They do contain a right of first refusal, prohibiting the distribution of derivative works. This right of first refusal supports the idea of a "one stop shop" for LRs.[14]

The commercial licences have a reasonable licence fee and do not limit the distribution of derivative works. The main reason for having a licence fee for commercial licences is that we do not want to disturb the existing commercial market for Dutch LRs, however small, by making our (government-funded) LRs available for free. The fee can be settled with a one-time lump sum or with a royalty scheme over a period of time. The former has the benefit of a reduced administrative burden (one single payment once and for all), while the latter requires less money to be put on the table upfront but implies an administrative follow-up process on potential revenues.

Within the framework of the STEVIN programme, a Pricing Committee was set up to advise the NTU and HLT Agency on LR pricing and licensing matters. The nine members of this committee come from Dutch and Flemish companies, funding bodies, government organisations and research institutes and were selected because of their expertise in business, open innovation, valorisation and technology transfer.

For LRs that are acquired outside of the STEVIN programme, certain procedures are different. The transfer of IPR to the NTU, for example, is not obligatory; the rights to the LRs remain with the developers. Furthermore, open source is a standard licence model for the HLT Agency. This was not the case with STEVIN. Evaluation and validation are conducted by the HLT Agency, unless they are already part of the LR's production process.

[14]Note that STEVIN project consortium members "automatically" receive a distribution licence when they, as suppliers, transfer their foreground knowledge to the NTU. This allows them to continue their work using "their" project results.

21.3.5 Services

The support that the HLT Agency provides to the LRs is based on knowledge management. Knowledge management is important for at least two reasons. Firstly, the availability of the knowledge does not stay limited to the availability of the expert(s) and secondly, once collected, the knowledge can easily be used, shared, kept up-to-date and expanded. The HLT Agency ensures that LRs and all knowledge about them are made and kept available.

21.3.5.1 Sources of Knowledge

For the HLT Agency there are three primary sources of knowledge: knowledge from external experts, knowledge collected by the HLT Agency while working with or maintaining an LR and knowledge gained by the service desk through question answering.

The first source of knowledge is made available to the HLT Agency in the form of documentation and is also the result of meeting with the project team at the end of projects. Most LRs come with accompanying user and technical documentation. In addition, a considerable amount of information can usually still be obtained from the STEVIN project that created the LR, e.g. in the form of progress reports or a project wiki. The HLT Agency asks the project teams to make this information available as an additional valuable source of background knowledge. When a new LR is supplied to the HLT Agency, a knowledge transfer meeting is held with the project team. In some cases we ask the experts to explain in detail how certain parts of the LR came into existence. For example, we interviewed the lexicon expert of the Spoken Dutch Corpus project and recreated the workflow for deriving the accompanying lexicon from the corpus, which would not have been possible on the basis of the documentation alone.

Secondly, the HLT Agency creates knowledge about LRs while using and maintaining the LR. Often the user manuals of software resources created by research projects do not provide a detailed description of all possible functions. Some functionalities may not be documented at all, or they are hard to find in the user interface. Studying user manuals and software has already resulted in additional knowledge and several improved user manuals and user interfaces. Besides, a lot of knowledge is gained while maintaining LRs: working on new versions greatly improves our understanding and knowledge of LRs.

The third main source of knowledge for the HLT Agency is the question-answering provided by the service desk. The service desk is more than simply a help desk, because, a.o., it processes orders and grants access to LRs. Answers to questions are also stored and made available for reuse in case similar questions are asked. The HLT Agency has agreements with the providers or external experts regarding question-answering: when questions require knowledge that the service desk does not (yet) have, the question is forwarded to the expert. The answer provided by this expert is forwarded (with acknowledgements) and stored by the

service desk. By following this procedure, the expert does not have to come up with the same answer to the same question over and over again and the HLT Agency keeps expanding its knowledge reservoir. The service desk thus acts as a filter, reducing the amount of repetitive questions to be answered by the experts, while they are actually given credit when an answer is reused.

21.3.5.2 An Integrated Knowledge Management Cycle

Knowledge management activities start when LRs are being created: formats and standards are discussed with project teams and intermediate versions of LRs are distributed. A crucial phase in knowledge management is the moment of transition: when LRs are handed over to the HLT Agency, as much (finalised) information as possible is collected and a knowledge transfer meeting with the project team is requested. The resulting knowledge is stored in e.g. wikis (for collaborative, online work) and in documents on our servers (for finalised information). The new LR is added to the service desk and web shop and this entire process is tracked in a workflow system. Knowledge management does not end here: while managing the lifecycle of the LRs, personal and documented knowledge is updated and any new knowledge is added, for example generated in the process of LR maintenance, or resulting from answering questions through our service desk. The HLT Agency also keeps an overview of who uses the LRs for what purposes, which supports marketing efforts and helps to bring users together.

21.3.5.3 User Support

General HLT support is provided on request. Depending on the amount of work required, the HLT Agency either offers this service for free, or charges a small fee, or applies an hourly rate. Examples of this type of support are: (a) helping researchers choose appropriate standards for their data collection, (b) connecting users to organisations which can provide in their specific needs, and (c) automatically tagging data sets for others (who are not willing or able to install and use the required tools). Also other actions, useful for users and suppliers, are undertaken. E.g., the Dutch Parallel Corpus was the first LR to receive an ISBN/EAN number issued by the INL, which will facilitate referencing and citation and will improve the LR's visibility. The idea is to provide every (STEVIN) corpus with an ISBN/EAN number.

21.4 Target Groups and Users

Researchers from various disciplines turn to the HLT Agency to access all sorts of LRs, such as general, socio-, computational and forensic linguistics, translation studies, social studies, cognition studies, historical and bible studies, communication

and information studies, and Dutch studies from all over the world. Before the HLT Agency existed, researchers often had to collect their own LRs before being able to start their research proper. The advantages of this new approach in which LRs are made publicly available for researchers cannot be overestimated. For instance, since researchers do not need to allocate time and money for data collection, they can start their investigations earlier and devote more time to research. In addition, they can base their investigations on data collections that are officially documented and traceable. This is important for reviewing and replication purposes and is in line with new trends favouring open access.

Teachers and students can also access LRs for educational purposes. Frequency lists were used as a starting point in second language education or implemented in educational applications for specific groups, such as dyslectics. Audio has been used in e.g., educational games and quizzes.

Small and medium enterprises (SMEs) are another important target group for the HLT Agency. SMEs are often willing to develop useful HLT applications, but they are not always able to bear the costs of developing the LRs that are required for such applications. The availability of LRs at affordable prices through the HLT Agency lowers cost barriers and offers a viable solution. Take for example a small company that provides speech solutions for a specific user group like people with reading difficulties. The HLT Agency can offer reference lexicons, or a part of a lexicon, at a reduced price, for improving the company's speech synthesis system. The HLT Agency can also provide support or advise, based on knowledge of the LRs.

In addition to these specific target groups, a wide variety of users turn to the HLT Agency for LRs, such as lawyers, language amateurs and even artists. Examples of their use of LRs are the use of a speech corpus in a court case (where a telephone recording had to be linked to a certain person and a Dutch language model had to be constructed), the use of lexical data by crossword enthusiasts, a Dutch family abroad who wanted to teach their children Dutch, and the work on an art object incorporating speech from the Spoken Dutch Corpus.

21.5 Challenges Beyond STEVIN

With the end of the STEVIN programme, the steady and guaranteed in-flow of new LRs comes to an end. The fact that also non STEVIN projects are handing over their LRs to the HLT Agency illustrates the importance of the HTL Agency for the HLT field in Flanders and the Netherlands. Nevertheless, for the future, the NTU (as principal and funder) and the HLT Agency (as agent) have to rethink and adapt their current policies and procedures to manage the LR lifecycle, in particular regarding:

- Criteria to select which LRs are to be acquired;
- A rationale to determine the most efficient and effective manner to make LRs available;

- Guidelines to determine when LRs need which form of maintenance;
- Procedures to establish whether and which new LRs are required for Dutch;
- Strategies to raise awareness of LR availability and potential, also for companies;
- Ways to organise "knowledge platforms" and communities of practice centered around specific LRs.

One dominating element in the overall policy remains the continuous support for the Dutch language in general. Hence, the NTU may choose to favour the support (or development) of LRs that do not have a high commercial value or a large number of potential users, but which might present a high "value" for specific target group(s) of the NTU. Such LRs would simply not become available for reuse if organisations like the HLT Agency did not accept them. This seemingly resembles the Long Tail strategy: "selling a large number of unique items with relatively small quantities sold of each" [1]. However, one should not forget that this only applies for the storage and distribution aspects (via the web store) in the LR lifecycle, which are, relatively speaking, straightforward and low cost activities. Clearing the IPR, performing maintenance, and managing and expanding knowledge concerning an LR are complex and time intensive (and thus costly) activities. Hence, a judicious choice must be made about which LRs are worth spending the relatively scarce time and (public) money on (they constitute the "short tail"). The experiences gained so far and the available external expertise (e.g., the Pricing Committee – cf. Sect. 21.3.4) the HLT Agency can tap in, provide a solid basis to successfully tackle these challenges. The fact that the new South African National HLT Resource Management Agency [7] preferred a collaboration with the HLT Agency illustrates the soundness of the HLT Agency's operating model and the professionalism of its collaborators.

Another item to consider is how the HLT Agency will position itself with respect to emerging networks, such as CLARIN and META-NET, which also aim at taking care of (parts of) the LR lifecycle. For this purpose, the NTU has become a member of the CLARIN ERIC.[15] The HLT Agency already is an active partner in CLARIN-NL, the Dutch national branch of CLARIN, and will integrate into the CLARIN infrastructure as many digital Dutch LRs as possible. At the time of writing, the CLARIN-ERIC just started so that the real challenge here has only begun.

21.6 Conclusions and Future Perspectives

Thanks to STEVIN, the HLT Agency has become a linchpin of the Dutch-Flemish HLT community. Since its inception in 2004, the HLT Agency has gradually gained recognition in the HLT community in the Netherlands, Flanders and abroad. The idea of a central repository for (digital Dutch) LRs is widely supported and has

[15]The CLARIN-ERIC is the permanent management structure governing the CLARIN-network.

been taken up internationally. The STEVIN resources are important building blocks for the digital Dutch language infrastructure. While the STEVIN programme comes to an end in 2012, the HLT Agency will continue to act as a manager, maintainer, distributor and service desk for these and other LRs. After 7 years of mainly accompanying programmes and projects that produce LRs, the time has come to focus on the use and valorisation of their results.

In addition to STEVIN, the NTU and the INL, other parties are depositing their LRs at the HLT Agency. The sustainability of LRs is supported by adopting a clear licensing, pricing and IPR policy, maintaining the LRs, actively managing knowledge about the LRs and providing a service desk for question-answering. Although the policy and procedures adopted are subject to change over time, the goals of making and keeping digital Dutch LRs available to strengthen the position of the Dutch language in today's information society will be pursued in the future too.

In short, the HLT Agency will ensure that digital Dutch LRs, especially those derived from the STEVIN research programme, will continue to have their lifecycles properly managed and will be optimally available for research, education and commercial purposes. As of 2013, the HLT Agency is no longer hosted by the INL, but integrated in the NTU. New contact details are www.tst-centrale.org and servicedesk@hlt-agency.org.

Acknowledgements We thank the three anonymous reviewers, Anna Aalstein and Boukje Verheij for their valuable comments on earlier versions of this text.

References

1. Anderson, C.: The Long Tail: Why the Future of Business is Selling Less of More. Hyperion, New York (2006)
2. Beeken, J.C., van der Kamp, P.: The centre for Dutch language and speech technology (TST Centre). In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC04), Lisbon, pp. 555–558 (2004)
3. Binnenpoorte, D., Cucchiarini, C., D'Halleweyn, E., Sturm, J., de Vriend, F.: Towards a roadmap for human language technologies: the Dutch-Flemish experience. In: Proceedings of the 3th International Conference on Language Resources and Evaluation (LREC02), La Valletta (2002)
4. Boekestein, M., Depoorter, G., van Veenendaal, R.: Functioning of the centre for Dutch language and speech technology. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC06), La Valletta, pp. 2303–2306 (2006)
5. Choukri, K., Piperidis, S., Tsiavos, P., Weitzmann, J-H.: META-SHARE: Licenses, Legal, IPR and Licensing issues. META-NET Deliverable D6.1.1 (2011)

6. Cucchiarini, C., Daelemans, W., Strik, H.: Strengthening the Dutch language and speech technology infrastructure. In: Notes from the Cocosda Workshop 2001, Aalborg, pp. 110–113 (2001)
7. Grover, A.S., Nieman, A., van Huyssteen G., Roux, J.: Aspects of a legal framework for language resource management. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC12), Istanbul, pp. 1035–1039 (2012)
8. Mariani, J., Choukri, K., Piperidis, S.: META-SHARE: Constitution, Business Model, Business Plan. META-NET Deliverable D6.3.1 (2001)
9. Oksanen, V., Lindén, K., Westerlund, H.: Laundry symbols and license management: practical considerations for the distribution of LRs based on experiences from CLARIN. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC10), La Valletta (2010)
10. Oostdijk, N.: Chap. The design of the spoken Dutch corpus. New Frontiers of Corpus Research, pp. 105–112. Rodopi, Amsterdam (2002)
11. Pogson, G.: Language technology for a mid-sized language, part I. Multiling. Comput. Technol. **16**(6), 43–48 (2005a)
12. Pogson, G.: Language technology for a mid-sized language, part II. Multiling. Comput. Technol. **16**(7), 29–34 (2005b)
13. Spyns, P., D'Halleweyn, E., Cucchiarini, C.: The Dutch-Flemish comprehensive approach to HLT stimulation and innovation: STEVIN, HLT Agency and beyond. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC08), La Valletta, pp. 1511–1517 (2008)

Chapter 22
Conclusions and Outlook to the Future

Jan Odijk

22.1 Introduction

The preceding chapters have sketched the context of the STEVIN programme and detailed descriptions of some of the results of the scientific projects carried out in the STEVIN programme. In this chapter I will briefly and very globally describe the impact of the STEVIN programme as a whole to human language technology (HLT) for the Dutch language in the Low Countries (cf. Sect. 22.2). In Sect. 22.3 I will identify a number of research data and topics that are, despite the STEVIN programme, still insufficiently covered but needed. Here I will also take into account international developments in the field of HLT that are relevant in this context. In Sect. 22.4, I identify recent international trends with regard to HLT programmes, assess the position of the Dutch language, and the prospects for funding of programmes and projects that are natural successors to the STEVIN programme. I also make some suggestions to government administrations for future policy actions. In Sect. 22.5, I summarise the major conclusions of this chapter.

22.2 Results of the STEVIN Programme

In this section we briefly and globally discuss the results of the STEVIN programme and its impact on HLT for the Dutch language in the Low Countries. More details about the results of the STEVIN programme can be found in Chap. 1, page 1.

It is clear from this book that the main objectives of the STEVIN programme have been achieved. Firstly, an effective digital language infrastructure for Dutch,

J. Odijk (✉)
UiL-OTS, Trans 10, 3512 JK Utrecht, The Netherlands
e-mail: j.odijk@uu.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_22, © The Author(s) 2013

based on the BaTaVo priorities[1] has been realised. An impressive amount of properly documented language resources (both data and software) has been created in the STEVIN programme, together with guidelines, conventions, and best practices for creating these and similar language resources. These language resources and their documentation are stored and maintained at the HLT Agency, and made available to all researchers and developers in a non-discriminative way, either via resource-specific licenses or under an Open Source License (cf. Chap. 21, page 381). The intellectual property rights (IPR) surrounding the language resources have been explicitly dealt with, so that the resources can actually be used by researchers and developers.

Second, strategic HLT research has been carried out in the STEVIN programme in a range of projects, selected, again, on the basis of the BaTaVo priorities.

Thirdly, the STEVIN programme has enormously stimulated network creation among players in HLT. There has been close cooperation between researchers from academia and industrial developers. There has also been close cooperation between partners from the Netherlands and partners from Flanders.[2] The STEVIN programme has consolidated the HLT activities in the Low Countries. It has created and supported a wide range of events where researchers, industrial developers, policy makers, potential users from industry and governments could meet, exchange ideas, and discuss important technical and policy issues. The STEVIN programme has educated new experts:

- Recently graduated students and post-docs as a side effect of the research and resource creation projects they were participating in;
- Students and potential students via educational activities;
- Policy makers and decision makers from government and companies via master classes.

STEVIN has also contributed significantly to the transfer of knowledge from academia to industry and vice-versa through the various projects in most of which academia and industry collaborated, as well as via several publications on the STEVIN programme that have appeared in journals published by ministries and the journal of the NOTaS[3] organisation DIXIT.[4]

The STEVIN work programme lists a number of potential (classes of) HLT applications as illustrations. The STEVIN projects have contributed to the realisation of such applications in many ways: in some cases indirectly, e.g. by creating resources required for the development of the technology underlying the application or by

[1] The STEVIN priorities have been listed in Chap. 1, Table 1.1 (page 2) and have been derived from the BaTaVo priorities [7]. 'BaTaVo' is an acronym standing for **Ba**sis**Ta**al**Vo**orzieningen (Basic Language Resources).

[2] Over 330 binary cooperation link occurrences in the STEVIN projects alone witness to the extent of collaboration in the STEVIN programme.

[3] NOTaS is a professional organisation for HLT in the Netherlands. See http://www.notas.nl/

[4] http://www.notas.nl/en/dixit.html

doing strategic research on underlying technologies; in other cases more directly by carrying out application oriented research. Some projects dealt with the creation of the application itself, e.g. in demonstration projects. Multiple STEVIN projects have contributed to the priority example applications, which were: information extraction from speech, detection of accent and identity of speakers, extraction of information from (monolingual or multilingual) text, semantic web, automatic summarisation and text generation, automatic translation, and educational systems.

The STEVIN programme has been evaluated halfway and at the end of the programme by external experts. In both cases the evaluations were very positive [13, 15].

Summarising, it can be concluded that the STEVIN programme has been very successful and has largely achieved its objectives in an excellent way.

22.3 Desiderata for the Near Future

The STEVIN programma has largely achieved its objectives, as described in the preceding section. However, this does not mean that the field of HLT for Dutch is now fully covered.

Firstly, there are a number of topics that have not been covered at all or only to a limited degree within STEVIN. This includes corpora for and research into multimedia, and corpora for and research into speech synthesis. For these areas, this was in part intentional. STEVIN attempted to avoid overlap with the concurrently running IMIX programme,[5] which covers some aspects of multimedia. But even with IMIX, resources for multimedia are largely absent and research into it has been very limited.[6] Research into speech synthesis was considered not so useful because the state-of-the-art systems at the time were closed commercial systems.[7] Semantic analysis and semantic corpora were part of STEVIN (inter alia in D-Coi and SoNaR, cf. Chap. 13, page 219), but were represented there only to a very small degree. The lexical semantic database Cornetto (cf. Chap. 10, page 165) created in the STEVIN programme is evidently relevant for applications related to the semantic web, but no application-oriented research project had the semantic web as its focus. Annotation of discourse and rhetoric relations was not completely absent in STEVIN, but was not addressed in a systematic manner or on a sufficiently large scale. Morphological analysis of derivation and compounding is lacking completely. Corpora for and research into robust speech recognition were well represented in STEVIN, but, of course, not all problems of robust speech recognition are solved with it. There

[5] http://www.nwo.nl/imix

[6] Some (Dutch) organisations (TNO, Twente) were involved in the European FP6 projects AMI and AMIDA, which deal with multimedia. See http://www.amiproject.org/

[7] The work done in the Autonomata project (Chap. 4, page 61), however, is surely relevant to speech synthesis.

are many situations were adverse conditions make automatic speech recognition a challenge, and only some of them were addressed in STEVIN. And this also holds for other research areas. Because of limitations of budget and time, many of these areas could be covered only in part. The large number of excellent project proposals (that would qualify for funding if there were enough money) witness to the fact that there are still many areas in which excellent research can be carried out that unfortunately could not take place in the STEVIN programme.

Secondly, STEVIN has, by its success, yielded new data that enable researchers to address existing research questions in a better way (e.g. in a more data-intensive way). STEVIN also made it possible to address completely new research questions. Though many of the newly created insights, data and software have already been used (sometimes in preliminary versions) by other projects in the STEVIN programme, the complete data sets of e.g. SoNaR (Chap. 13, page 219) and Lassy (Chap. 9, page 147) have become available only in a rather late stage in the programme. Therefore, the potential that they create for research in HLT as well as for other fields (e.g. research in various subdisciplines of linguistics) has hardly been exploited. A critical remark on the research carried out in STEVIN has been that it was largely incremental in nature, and that in some projects the research replicated earlier research but now for Dutch [6]. This may well be true, and is not unexpected given the nature of the STEVIN programme. But STEVIN has prepared the grounds for ground-breaking and cutting edge research in HLT, so that new research programmes and projects are now needed to tap into this potential and optimally exploit the rich resources of data and software that STEVIN has yielded.

Thirdly, significant developments on an international level have occurred as well. Firstly, IBM has shown, with its *DeepQA* approach, that language technology can be used to robustly extract precise answers to specific questions from a mix of structured (e.g. databases) and unstructured data (e.g. texts on the world wide web).[8] Though IBM demonstrated the capabilities of the Watson DeepQA in a game context (the Jeopardy! Game), it is now setting up systems based on the same principles for commercial applications (for example in the medical domain). With this, IBM has set the extraction of precise answers from unstructured data central on the HLT agenda. Analysis of unstructured documents is also needed in the context of business intelligence, e.g. in determining the perception of a company's public image and of specific commercial products and services by clients and prospects on the basis of an analysis of unstructured text resources in modern social media such as blogs, product review web sites, Twitter, etc. This topic is of great importance for big companies such as Philips and SAP [12]. They currently carry out such analysises still largely manually, but that is becoming infeasible with the exponentially growing digital content. The same techniques are also needed and increasingly demanded in the area of humanities research: the sharp increase in the number of available digital documents and digital audiovisual material makes it impossible to work in the traditional manner. The analysis of large document collections and audiovisual

[8] http://www.research.ibm.com/deepqa/deepqa.shtml

materials requires the use of sophisticated HLT as auxiliary tools for research into linguistics, literature, history, culture, and political sciences. In the area of speech dialogue, Apple's SIRI on iPhone[9] has brought speech dialogues to a new level. This is only possible thanks to two factors: firstly, the speech recogniser contained in it is extremely robust to background noise of a large variety of environments, to make its use on a mobile phone feasible. Secondly, SIRI maximally uses contextual information (largely available on the owner's iPhone) to interpret (usually ambiguous and underspecified) utterances spoken by the phone owner in the dialogue.

STEVIN has not specifically addressed the issue of Question Answering on the basis of unstructured data, though it has prepared many components (e.g. the tools used and further developed to create the SoNaR and Lassy corpora; some components developed in the DAISY (Chap. 19, page 339) and DAESO (Chap. 8, page 129) projects). Within STEVIN there was some research in the area of opinion mining (in the DuOMAn project, Chap. 20, page 359), but the large industrial interest justifies investing more in this area. Robust speech recognition in adverse conditions has explicitly been addressed in STEVIN, but as stated above, surely not exhausted. Research into spoken dialogue and a maximal use of context in interpreting the utterances and guiding the dialogue has been largely absent in STEVIN, but surely deserves more attention.

In short, with the STEVIN programme finished, the Dutch and Flemish HLT researchers are in an excellent position to deepen the research and to extend it to new areas, some of which are of big importance to industry and other scientific areas such as the humanities. They look forward to new opportunities to maximally exploit the insights gained and the materials created in STEVIN in new research programmes and projects.

22.4 Future

Unfortunately, the prospects for funding a successor project to STEVIN are grim. In the Netherlands, consultation meetings with the NOTaS organisation have been held,[10] and in Flanders a round table meeting with some 40 players from the field [11]. A policy document sketching the outlines of a research programme in the area of the extraction of information from unstructured data (both textual and audiovisual) is available [2]. But obtaining funding for such a research programme is not easy. There is firstly the fact that the funding opportunities have to come both from the Netherlands and from Flanders. However, the priorities of the two governments differ, the instruments are different and the timing is difficult to synchronise. Furthermore, in the Netherlands there are no opportunities for

[9] http://www.apple.com/iphone/features/siri.html

[10] See e.g. http://taalunieversum.org/taal/technologie/taalinbedrijf/documenten/notas.pdf

discipline-specific research programmes. All research is organised via what are called *Top Sectors*, a series of sectors identified by the Dutch government as specifically important and with high potential for the Dutch economy. HLT research fits in very well in the Creative Industry Top Sector [14, 16]. For example, HLT can obviously be used in many applications in the areas of social media, publishers, TV and radio, gaming, museums and cultural heritage, and in mobile applications. But any research programme in this top sector will cover multiple disciplines in which HLT must try to find its place. In Flanders, the situation is different but [11] also concludes that opportunities must be sought to embed HLT research in broader initiatives. One possibility is to focus research and development effort around an integrated demonstrator that requires research into and development of technology from multiple disciplines, HLT being one of them.

In order to improve the chances of obtaining funding, the visibility and strength of the HLT sector can and must be further improved. Some HLT organisations in the Netherlands are united in NOTaS but certainly not all. Though CLIF[11] unites the scientific HLT community in Flanders, [11] argues in favour of the creation of a structure that unites and promotes the whole HLT sector.[12]

On the other hand, there are other developments that are directly relevant and make one more optimistic.

Firstly, some researchers from linguistics and HLT, in particular from the Netherlands, have, for several years now, been arguing for the need of setting up a distributed technical research infrastructure for humanities research. These efforts have led to a proposal for such an infrastructure called CLARIN (Common Language Resources and Technology Infrastructure). CLARIN has been put on the ESFRI roadmap in 2006. The ESFRI-funded European CLARIN preparatory project has been successfully executed.[13] CLARIN has been put on the national Dutch roadmap for large infrastructures in 2008. Since 2009 the national research infrastructure project CLARIN-NL[14] is running. The targeted research infrastructure will contain, inter alia, a range of HLT data and tools. These data and tools must be adapted to make them user friendly and easy to use for humanities researchers without an HLT background. Though CLARIN in Flanders has so far only been awarded funding for preparatory activities, a modest budget from Flanders could be secured for cooperation with the Netherlands. Together with funding from the CLARIN-NL project, a small-scale cooperation project between the Netherlands and Flanders could be set up to make the tools developed in the STEVIN programme (especially the ones created or extended in the D-Coi, SoNaR and Lassy projects) cooperate seamlessly with each other as web services in a work flow system. This project, called TTNWW, runs from 2010 to 2012.[15] A decision on larger scale

[11] Computational Linguistics in Flanders, http://clif.esat.kuleuven.be/

[12] The need for this is also felt at the European level, see below.

[13] http://www.clarin.eu

[14] http://www.clarin.nl

[15] More information on this project can be found here: http://www.clarin.nl/node/76#TTNWW

activities for CLARIN in Flanders is expected in the course of 2012. If that decision would be positive, it might open up new opportunities for collaboration between the Netherlands and Flanders in HLT, though the focus will be on applying HLT in a research infrastructure for humanities researchers, but not on research into HLT itself.

Secondly, a range of projects is working on the research agenda for HLT at the European level. For example, the FLaReNet project[16] has consulted the HLT community in Europe and beyond to formulate recommendations for the policy of the European Union with regard to language resources. It has resulted, inter alia, in the *FLaReNet Strategic Language Resource Agenda* [5] and the FLaReNet recommendations for language resources [4].

Of particular importance in this respect is the fact that the European Union keeps expanding, and is becoming increasingly more multilingual. The multilinguality of Europe is on the one hand considered a valuable cultural asset. On the other hand it also is a burden because it makes communication more difficult and especially more costly. The European Commission is seeking ways of reducing these costs. HLT has the potential to significantly reduce the costs for Europe's multilinguality and even to turn this into an economic asset. Google has already proved that large-scale machine translation for several tens of language pairs is feasible for certain applications and services where the translation quality is of secondary importance.[17] Several policy makers in the European Commission believe that the HLT community in Europe has the potential to improve machine translation significantly, so that it becomes useful for applications where translation quality does matter. However, this is only possible if the research community is united, joins forces in a common strategic research agenda, and receives sufficient means to carry out ground-breaking research. These are pre-conditions for achieving the goals set and to avoid dependency on foreign commercial companies. Several projects in the EU ICT programme have been started up (arguably in part as a result of the FLaReNet project) to work towards such a situation, which hopefully can become part of Europe's Horizon 2020 Programme. These projects include META-NET[18] and various related projects such as META-NORD,[19] CESAR,[20] and METANET4U.[21] The META-NET project is carrying out some research, in particular it is building bridges to relevant neighbouring technology fields via its META-RESEARCH activities. But even more importantly, the META-VISION part of the project has consulted a wide range of players in the field, including researchers, commercial and non-commercial HLT developers, commercial users of HLT, language professionals, and others, to inventory needed and desired functionality, important research areas, commercial

[16] http://www.flarenet.eu/

[17] http://translate.google.com/

[18] http://www.meta-net.eu/

[19] http://www.meta-nord.eu/

[20] http://www.cesarproject.eu/

[21] http://metanet4u.eu/

potential for HLT, etc., to develop a vision on the future of HLT for the next decade. This vision has been created and laid down in a vision paper [8]. Currently, this vision is being developed into a strategic research agenda. At the same time, META-NET has assessed the status of HLT for the European Union languages, and it has described properties of each individual language that pose specific challenges to HLT for that language. This has resulted in an impressive range of *language white papers*, preliminary versions of which are already available,[22] including one for Dutch [9]. Not surprisingly, the Dutch language scores very well here, and plays in the same league as big European languages such as French and German. For a large part, the STEVIN programme is to be credited for this.[23]

The META-NET project is also working on improving the pre-conditions for carrying out excellent research and efficient technology development. In particular it aims at facilitating the sharing and exchange of language resources via the open distributed META-SHARE language resource exchange facility [10].[24] Obviously, though META-SHARE has a different goal and a different target group, there are commonalities with the CLARIN infrastructure, there is close collaboration between the two projects, and shared use of certain technology (e.g. both make use of the CMDI framework for metadata[25]). In this context it is also important to see how the status of the Dutch HLT Agency is developing. It is natural that it would develop into a data centre not only in the CLARIN infrastructure (which it is already working towards) but also in META-SHARE.

In these European developments, one can discern a parallel with the developments in the Netherlands and Flanders 10 years ago: the language white papers can be considered as the equivalent of the BaTaVo report [7] for the Dutch language, but now for all languages of Europe and with a special focus on multi- and cross-linguality; the META-SHARE facility that is being prepared can be considered the equivalent of the HLT Agency, though again now on a European scale; and the META-NET vision paper and the strategic research agenda that is being developed correspond to the policy documents made in the Netherlands (e.g. [1]) that have contributed to the positive decision to fund the STEVIN programme.[26]

It remains to be seen whether these efforts will indeed lead to a common strategic research agenda and sufficient funding to carry out the research and development that will be necessary to execute this research agenda. However, the fact that these efforts are being made turn one optimistic in believing that it will create

[22] http://www.meta-net.eu/whitepapers

[23] The final versions of the language white papers will be come available Mid 2012.

[24] http://www.meta-net.eu/meta-share

[25] Component-based MetaData Infrastructure [3].

[26] Though these parallels are real, I do not intend to claim that the European developments have been inspired completely by the developments in the Netherlands. Other projects, e.g. Euromap (http://www.2020-horizon.com/EUROPEAN-OPPORTUNITY-MAPPING-(EUROMAP)-s50899.html), have undertaken similar activities at the European level already more than 10 years ago, for the situation at that time.

opportunities for European HLT researchers in general, and Dutch and Flemish researchers in particular.

22.5 Concluding Remarks

I briefly summarise the current situation as sketched in this chapter. It is very unlikely that there will be a successor programme that is similar in nature to the STEVIN programme. At the moment, there are also no concrete opportunities for such a programme. Nevertheless, there is at least one concrete example where funding has been obtained for an (admittedly small-scale) successor project (CLARIN TTNWW). In addition, there are many opportunities for carrying out research in HLT, not only in the Netherlands and Flanders (separately), but also at the European level. However, these are mainly opportunities for individual projects, not for programmes. The future will tell whether these opportunities are real and will materialise into concrete cutting edge HLT research projects.

Acknowledgements I would like to thank Peter Spyns and an anonymous reviewer for valuable comments on an earlier version of this chapter.

References

1. Akkermans, J., van Berkel, B., Frowein, C., van Groos, L., Compernolle, D.V.: Technologieverkenning Nederlandstalige taal- en spraaktechnologie. Report, Ministry of Economic Affairs, The Hague (2004)
2. Boves, L.: Enterprise language processing: Een aanzet voor een nieuw programma. Report, Nederlandse Taalunie, The Hague (2011)
3. Broeder, D., Kemps-Snijders, M., Uytvanck, D.V., Windhouwer, M., Withers, P., Wittenburg, P., Zinn, C.: A data category registry- and component-based metadata framework. In: Calzolari, N., Maegaard, B., Mariani, J., Odijk, J., Choukri, K., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), pp. 43–47. European Language Resources Association (ELRA), Valetta (2010)
4. Calzolari, N., Bel, N., Choukri, K., Mariani, J., Monachini, M., Odijk, J., Piperidis, S., Quochi, V., Soria, C.: Language Resources for the Future – The Future of Language Resources. CNR – Istituto di Linguistica Computazionale A. Zampolli, Pisa (2011). http://www.flarenet.eu/sites/default/files/FLaReNet_Book.pdf. FLaReNet Final Deliverable
5. Calzolari, N., Quochi, V., Soria, C.: The FLaReNet Strategic Language Resource Agenda. CNR – Istituto di Linguistica Computazionale A. Zampolli, Pisa, Italy (2011). http://www.flarenet.eu/sites/default/files/FLaReNet_Strategic_Language_Resource_Agenda.pdf

6. Daelemans, W.: What did STEVIN do for HLT research? http://taalunieversum.org/taal/technologie/stevin/documenten/stevin_results_r_28112011.pdf (2011). Presentation held at the STEVIN Final Event, Rotterdam, the Netherlands
7. Daelemans, W., Strik, H.: Het Nederlands in taal- en spraaktechnologie: prioriteiten voor basisvoorzieningen. een rapport in opdracht van de Taalunie. Report, Nederlandse Taalunie, The Hague (2002). http://taalunieversum.org/taal/technologie/docs/daelemans-strik.pdf
8. META-NET Consortium: The future European multilingual information society. Vision paper for a strategic research agenda. META-NET report, DFKI, Berlin (2011). http://www.meta-net.eu/vision/index_html/reports/meta-net-vision-paper.pdf
9. Odijk, J.: Languages in the European information society – Dutch (early release edition). META-NET white paper series, META-NET, Berlin (2011). http://www.meta-net.eu/whitepapers/download/meta-net-languagewhitepaper-dutch.pdf
10. Piperidis, S.: META-SHARE: An open resource exchange infrastructure for stimulating research and innovation. Presentation held at METAFORUM 2011, Budapest, Hungary (2011) http://www.mt-archive.info/META-FORUM-2011-Piperidis.pdf
11. Spyns, P.: TST-rondetafel 07/09/2011 – STEVIN-roadmapworkshop. Report, EWI, Brussels, Belgium (2012). Version 4 April 2012 http://http://taalunieversum.org/taal/technologie/stevin/vl_rondetafel/
12. Taal in Bedrijf: Panel on business intelligence. http://taalunieversum.org/taal/technologie/taalinbedrijf/programma_2011/ (2011)
13. Technopolis$_{|group|}$: Eindevaluatie STEVIN programma: Eindrapport. Report, Nederlandse Taalunie, The Hague (2011). http://taalunieversum.org/taal/technologie/stevin/documenten/stevin_eindevaluatierapport.pdf
14. TopTeam Creatieve Industrie: Creatieve industrie in topvorm: advies topteam creatieve industrie. Report, Ministry of Economic Affairs, Agriculture and Innovation, The Hague, the Netherlands (2011). http://www.rijksoverheid.nl/documenten-en-publicaties/rapporten/2011/06/17/creatieve-industrie-in-topvorm.html
15. Uszkoreit, H., Prószéky, G., Moore, R., Calzolari, N., Heisterkamp, P., Adda, G., Abeillé, A., Piperidis, S.: STEVIN mid term review. Report, Nederlandse Taalunie, The Hague (2008). http://taalunieversum.org/taal/technologie/stevin/documenten/iap_tussentijdse_evaluatie.pdf
16. van den Bosch, A., Odijk, J., Cucchiarini, C., van den Bosch, L.: Taal- en spraaktechnologie voor het Nederlands: toptechnologie voor de topsectoren. http://www.clarin.nl/system/files/TST-Topsectoren.pdf (2011)

Index

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6, © The Author(s) 2013