DUTCH SPEECH RECOGNITION IN MULTIMEDIA INFORMATION RETRIEVAL

Roeland Ordelman

Dissertation committee:

Prof. dr. F.M.G. de Jong, promotor
Dr. A. J. van Hessen, assistant-promotor
Prof. dr. ir. W.H.M. Zijm, chairman/secretary
Prof. dr. P.M.G. Apers
Prof. dr. T.W.C. Huibers
Dr. D.A. van Leeuwen (TNO Technische Menskunde, Soesterberg)
Prof. dr. ir. J.-P. Martens (Universiteit Gent)
Prof. dr. S. Renals (University of Edinburgh)

Taaluitgeverij Neslia Paniculata. Uitgeverij voor Lezers en Schrijvers van Talige Boeken, Nieuwe Schoolweg 28, 7514 CG Enschede.

CTIT Ph.D. Thesis Series No. 03-56. Centre for Telematics and Information Technology, University of Twente, P.O. Box 217, 7500 AE Enschede.

SIKS Dissertation Series No. 2003-12. The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

This research was funded by the following projects:

DRUID, a project within the scientific program of the Telematica Instituut.

ECHO, a project within the 5th framework program IST of the European Union.

Waterland, a technological co-operation project funded by the Dutch Ministry of Economic Affairs.

Printed by: Océ Facility Services, Enschede

ISBN 90-75296-08-8
ISSN 1381-3617; No. 03-56 (CTIT Ph.D. Thesis Series)

DUTCH SPEECH RECOGNITION
IN MULTIMEDIA INFORMATION RETRIEVAL

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 10 oktober 2003 om 13.15 uur

door

Roeland Jacobus Frederik Ordelman

geboren op 20 maart 1969
te Oud-Beijerland

Dit proefschrift is goedgekeurd door de promotor,
prof. dr. F.M.G. de Jong,
en door de assistent-promotor,
dr. A.J. van Hessen.

ACKNOWLEDGEMENTS

This thesis reports on research carried out at the Language, Knowledge and Interaction group (TKI) of the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente, the Netherlands. The presented research was funded in part by the Telematics Institute project DRUID, the European Union project ECHO and the Waterland project. I wish to acknowledge and thank those people who contributed to the completion of this thesis.

In particular, I thank my promotor Franciska de Jong for her support, feedback and patience. Irrespective of where she was, in the Netherlands, in Europe or elsewhere, I could expect an immediate response to any of my questions. Especially her ability to reveal phrasing weaknesses and unbalanced structures almost at a single glance, was of great help during the writing stage. In addition, I thank my daily supervisor Arjan van Hessen for sharing his expertise and knowledge. His unbridled enthusiasm and wide interests have been very stimulating in the more difficult times during the process, and a welcome diversion at work, privately and ’on the road’. I am very grateful for the help provided by David van Leeuwen at TNO Technische Menskunde, especially concerning the *ABBOT* speech recogniser and the UNIX operating system, and for providing most of the acoustic models that were used in this research. Without his support, this research would not have been possible. Special thanks are due to Paul Melis who contributed a great deal to the development of the grapheme-to-phoneme converter and has been of great help during the last stage of the research, enabling me to run a few experiments that I had already unwillingly labelled as ’future research’.

I thank my colleagues for their interest, productive discussions and (mental) support. I especially want to express my gratitude for the technical support provided by Hendri Hondorp and the B&O, technical support and maintenance group, in particular Fred Gansevles, Jan Veninga and Marc Berenschot. Without them keeping things going (or getting things going again), this research could not even have been started. Special thanks to Rieks op den Akker and Mannes Poel who read the manuscript and provided many valuable comments, Lynn Packwood for wielding the red marker concerning

the English language while reading the manuscript (and for finally convincing me to read those ... boring Manual Pages when I had just arrived at CS), my roommate Simon Keizer among others for sharing his tips&tricks, Charlotte Bijron and Alice Vissers of the secretariat for their care and invaluable work in the background, and last but not least, Anton Nijholt for being around backstage.

I would like to thank the following organisations:

- TNO Technische Menskunde, for the productive collaboration and for providing the broadcast news acoustic models,
- PCM publishers, for providing the daily feed of newspaper data for language model training,
- Van Dale Lexicografie, for providing the Van Dale pronunciation lexicon, and
- the Nederlandse Omroep Stichting (NOS), for providing the autocues data of the "NOS Acht uur journaal".

Finally, special thanks are due to my wife Anne. Her care, patience and encouragements have been of invaluable support.

Roeland Ordelman

Deventer, September 2003.

Contents

Part I

Multimedia Document Representation

Chapter 1

Introduction

This thesis addresses the development and application of Dutch speech recognition as a tool for searching Dutch multimedia content in a multimedia information retrieval environment. This chapter explains the role of speech recognition for the purpose of information retrieval and the motivation behind the research described in this thesis, summarised in the last sections of this chapter.

1.1 A representation mismatch

As data storage capacities grow to nearly unlimited sizes thanks to ever ongoing hardware and software improvements, we can preserve anything we want, given that it can be stored digitally. But assuming that the intention of data storage is to use (portions of) it some later time, the data must also be searchable in one way or another: it must be possible to formulate a particular *information need* related to the stored data and to *retrieve* this information from it.

With the huge quantities of data being stored today, efficient and successful retrieval is becoming more and more complicated. Given an information request, a search facility has to provide preferably all relevant items in the data collection (it should have a high *recall*), leaving aside non-relevant items (have a high *precision*) within an acceptable time limit. This requires sophisticated search algorithms that do not put too heavy demands on processing time. The research in *Information Retrieval* (IR), by Salton and McGill (1983) defined as the research “concerned with the representation, storage, organisation and accessing of information items”, aims at finding the optimal solutions for the fulfilment of these requirements.

It is not these huge data quantities alone that complicate IR research. As storage capacity is hardly a limiting factor anymore, all kinds of digital documents with different semiotic formats (texts, sounds, graphics, etc.), or even documents that combine different formats as in multimedia documents (typically videos), are being digitised and stored on a large scale as well. Searching non-text based types of data automatically, is far more complicated than searching the traditional text-oriented data types, basically because there is a *representation mismatch* that must be solved. This is depicted in Figure 1.1: a searching process can be described as trying to find a match between the information need, formulated in a *query* and represented in a query representation, and a collection of documents, represented in a document representation that is normally referred to as an *index*. This index is a compressed version of the documents capturing all important information contained in the documents. Using natural language to formulate an information need in a query, is from a user's perspective the most evident choice. With multimedia content, the representation of a natural language query does not match the representation of the documents (images in pixels, audio in samples). To enable automated searching in these documents, the representation of the query and of the document collection have to be put in agreement: by converting the document collection to the query representation (e.g., to text), by adjusting the query to the document representation (e.g., to images) or by converting both document and query representation to an intermediate representation (e.g., to sound units). Solving representation mismatches by converting document representations is one of the main issues in multimedia information retrieval.

1.2 Solving the representation mismatch

Circumventing a representation mismatch, some search engines on the Internet, such as *Google*, support the search for images by looking at the file names of images (such as "*computer.png*"), on the reasonable assumption that the file names reflect something of the content of the image. Anyone who has tried to find images on the Internet this way, may have noticed that this approach is not very successful. An approach reflecting a way of "looking" at the images themselves to see if they are relevant for a given query, requires the translation of the picture into a textual representation that can be matched with the text-based query representation. But as research in *image retrieval* has shown (see e.g., Eakins and Graham, 1999, for an extensive review), so called content-based image retrieval (CBIR) is extremely complicated. Therefore, the process is often reversed. Instead of adjusting the document to the query representation, the query is adjusted to the document representation: an example image acts as a query and the matching process is done at the image level (using for example similarity searches based on colour histograms). This type of searching is often referred to as "*query-by-example*" in the field of IR. However, one needs an example image first to find other images, which may impose a problem in itself.

Multimedia documents are often accompanied by some kind of textual information; at the least a file name or a title and maybe even a short de-

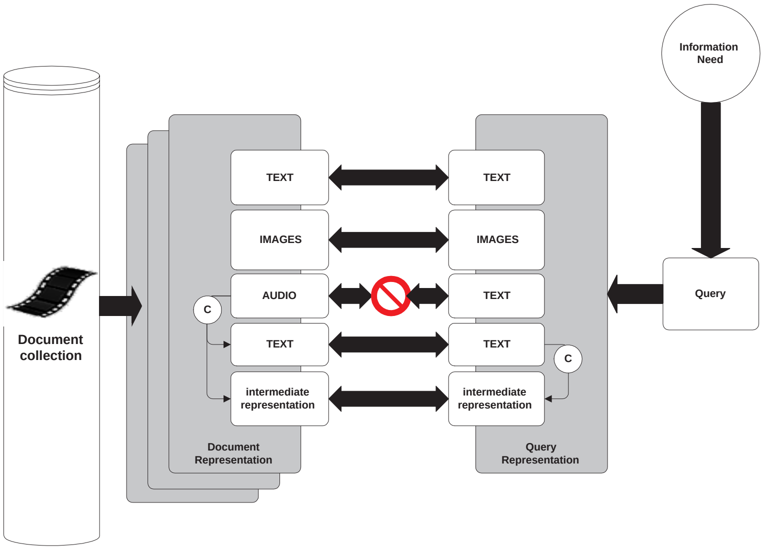

Figure 1.1: Document-query match/mismatch with multimedia data

scription of the item. Using this basic information as document representations can already be helpful in *locating* relevant documents. However, research in Multimedia Retrieval made clear that searching for specific pieces of information *within documents* as visualised in Figure 1.2—which is especially useful when the data is as unstructured as in multimedia data—requires a more detailed representation of the information contained in the documents. To enable direct access to relevant parts of the document, having time information available (where exactly can a piece of information be found in the document) is of crucial importance. It has already been noted that creating a detailed representation on the basis of images (video frames) is complicated. But although information contained in multimedia documents can be present in both audio (speech, music, sounds) and video (images), the audio alone can already provide useful information about the document's content and give clues about the presence of certain images or video fragments. Regarding *non-spoken audio*, for example the detection of the cheering of a crowd at a particular point in a video may indicate where a goal has been scored in a soccer match. Likewise, a particular tune can mark the start of a commercial or broadcast news show (List et al., 2001) and even music can carry useful information (Pfeiffer, 1999)[1]: for example,

[1] Note that music is not the primary retrieval source as is the case in Music Information Retrieval where for example melodies are sought in music (Goodrum and Rasmussen, 2000)

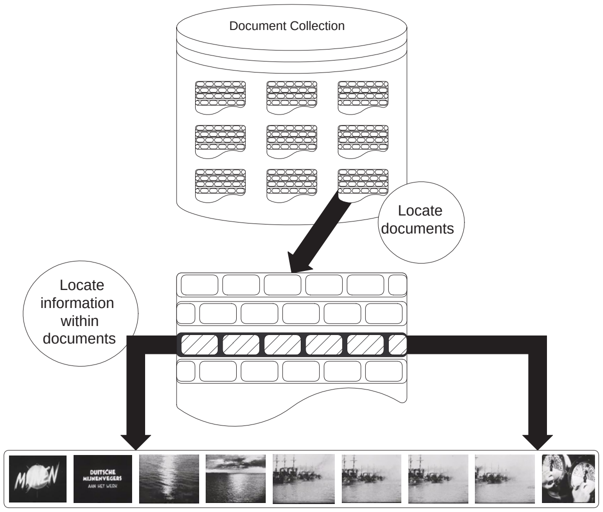

Figure 1.2: A more detailed representation of documents is required in order to access information *within* documents.

fast music usually accompanies video scenes with a lot of action, whereas slow music is typically used for love-scenes.

But undoubtedly the most informative component of the audio stream is the *spoken audio*. When at particular time slots in a video, the words "*Chagall*", "*exhibition*" and "*gallery*" are mentioned, it is likely that at that point a video fragment of the Chagall exhibition is included. By using *automatic speech recognition* (ASR) to convert existing speech into text, detailed textual multimedia document representations can be created. As speech recognition systems label recognised words with exact time information as a standard accessory, detailed searching within multimedia documents comes within reach: by deploying the time-codes produced by the ASR system, relevant video fragments can directly be accessed.

1.3 Spoken document retrieval

> ”Information is in the audio, video is for entertainment” - Richard Schwartz, BBN Technologies, Multimedia Retrieval Video-Conference, University of Twente, 1999

The retrieval of multimedia documents using the spoken audio parts is commonly referred to as *spoken document retrieval* (SDR) or alternatively, *speech retrieval*. SDR is an important research area within multimedia information retrieval (MMIR) research, aiming at the disclosure of multimedia documents. Strictly speaking, in this area the focus of SDR is not on spoken documents but on the spoken audio contained in multimedia documents. Although a wide range of MMIR applications in many different fields can be thought of—ranging from video mail retrieval (e.g., Brown et al., 1996) to systems for managing meetings[2], an illustrative and typical example of a field that can benefit from advances the research in MMIR and SDR is the broadcast sector. A typical application would be a search facility for the video archives to enable journalists to search for fragments that can be inserted in news items or documentaries. Broadcast companies produce streams of multimedia data on a daily basis and as there is a tendency to digitise as much as possible, more and more recordings are being stored in databases. Moreover, mainly for preservation reasons, the digitisation of older material and even historical archives is often in progress (retrospective digitisation).

To provide means for searching, such archives have traditionally been annotated with *human-generated metadata*. Metadata is usually defined as data *about* data. In the broadcast sector, *administrative metadata*, such as rights metadata (who is the legal owner of a video item) and technical metadata (such as the format of a video item), is an important metadata type, but *descriptive* human-generated metadata, including the title, duration, a short content description and a list of names and places that are mentioned in the video item, can also be preserved with the video items. Although the information density in this type of metadata is usually low, it can nevertheless be very helpful for locating specific video items in an archive. This type of information is therefore sometimes referred to as “bibliographic” or “tombstone information” among librarians[3]. However, locating specific parts or shots *within videos* remains time consuming: the usual way to find pieces of re-usable archive material, had always been scrolling manually through a large amount of data (sometimes literally a basket of video tapes) that matched some broader search criteria. If more

[2] See for example the M4 project: Multi-Modal Meeting Manager. Started on 1 March 2002, and is supported by the EU IST Program (project IST-2001-34485). URL: `http://www.dcs.shef.ac.uk/spandh/projects/m4/`

[3] Note that besides administrative and descriptive metadata, a third type of metadata is often distinguished: information about the structure and organisation of a multi-part digital object that can be encoded MPEG-7/21 (Goldman et al., 2003)

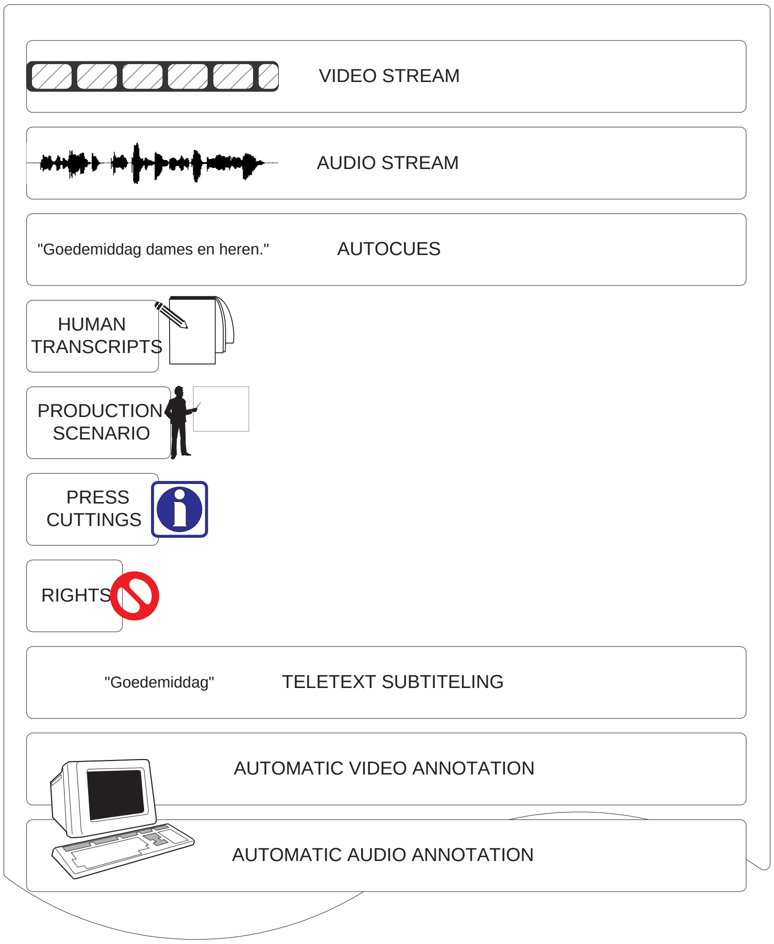

Figure 1.3: Illustration of the various metadata fields for a multimedia document: the video stream and the audio stream are the primary source of information. Autocues, human transcripts, production scenarios and automatically generated annotations of the video and audio material are examples of metadata that can be used for locating information

detailed information could be added to the metadata fields of the items in a multimedia archive, such as descriptions of the video shots and transcripts of spoken parts in the video[4] as depicted in Figure 1.3, the metadata would provide better means for locating interesting sections. As mentioned before, crucial additional information with respect to information located within documents is time information: where exactly in the documents can specific information be found. For cost efficiency reasons, adding such detailed information manually is not an option for huge amounts of data. In order to generate detailed multimedia metadata, researchers have been exploring other options. One of these is employing related text sources[5], such as the auto-cues of a broadcast news show, press cuttings, production scenarios and teletext subtitling.

But the greatest challenge in MMIR research is the exploration and development of a great variety of audio and image analysis tools to produce *automatically generated metadata* from the multimedia content. The most obvious examples are speech and image analysis tools but for an optimal performance of the these tools, additional pre-processing and post-processing tools (e.g., speech/non-speech detection in speech recognition) are indispensable. In addition, there is more information in the audio and video tracks than speech and images alone. With the appropriate tools, shot boundaries (video) or speaker (audio) changes for example, can be detected and can provide valuable information. Ultimately, the bundling of these tools should enable an automatic and detailed annotation of multimedia material, as if the material had really been “listened to” or “looked at.” Sometimes even adding information that manual annotations usually do not provide.

Many tools, alone or in combination, are already deployed effectively for the indexing of multimedia archives: well-known are the Carnegie Mellon’s *Informedia* project[6] (e.g., Witbrock and Hauptmann, 1998) and the *THISL* project (Robinson et al., 1999). The *BBN* system (Broadcast News Navigator, Maybury et al., 1997) and Virage[7] are examples of commercial systems that use ASR techniques for multimedia retrieval. In the DARPA[8] sponsored benchmark tests for video retrieval, such as performed in the Text REtrieval Conference (TREC), the speech recognition component in these multimedia retrieval environments has proven to be a most valuable tool (cf. Baan et al., 2002). A lot of useful information can be found in the spoken audio contained in a multimedia document and by deploying the state-of-the-art in current speech recognition technology, a considerable part of information can successfully be recovered.

[4] Transcriptions of spoken language recordings are sometimes interpreted as the fourth metadata type, next to administrative, descriptive and structural metadata

[5] When such text sources are not available in computer readable format, OCR techniques can be applied (see e.g., Harman and Voorhees, 1997))

[6] `http://www.informedia.cs.cmu.edu`

[7] `http://www.virage.com`

[8] Defense Advanced Research Projects Agency (DARPA)

1.4 Spoken document retrieval for Dutch

At the University of Twente a wide range of issues related to (multimedia) information retrieval have been studied (recent studies include Hiemstra, 2001; Petković, 2003; Velthausz, 1998; De Vries, 1999). However, at the start of the research described in this thesis, SDR had only sparsely been addressed. Some experience was obtained with SDR for English and French in the Olive project (De Jong et al., 1999). With the *DAS+* project, the exploration of SDR for the Dutch language had been started (Kraaij et al., 1998). Extending this research themes for Dutch content is necessary in order to keep up with the international technological advances and state-of-the-art in this field, to ensure that the increasing amount of information being stored in Dutch multimedia and spoken-word collections remains accessible. To prevent the huge amounts of Dutch data practically get lost as there are no adequate means for searching these, spoken document retrieval research for the Dutch language is regarded to be of vital importance both from an economical and historical perspective. Furthermore, in order to be able to participate in future collective research initiatives on an international level in the domain of spoken-word audio collections, such as recently promoted by the EU-US Spoken Word Archive Group (Goldman et al., 2003), expanding Dutch SDR research is of crucial importance.

The primary aim of the research described in this thesis, was therefore to bridge the technological gap between international state-of-the-art in SDR and the lack of experience of SDR for Dutch, and to explore the wide range of issues concerning spoken document retrieval, specifically for the Dutch language. Addressing Dutch SDR, was originally pushed by a number of multimedia retrieval projects, listed in Appendix A, aiming at the disclosure of a variety of multimedia collections. Solving the representation mismatch between the spoken audio parts in the collections and textual query representations, was regarded as indispensable for successful retrieval of the multimedia documents in these projects. The spoken document retrieval research described in this thesis took place in a multimedia information retrieval context, but it must be noted that the larger part of this research can be just as well interpreted with a focus on the (mono-media) spoken audio alone.

With Dutch SDR in mind, a suitable Dutch speech recognition system that could be used for the creation of suitable representations of the spoken audio contained in the multimedia documents, had to be identified or developed. As the characteristics of the task domain highly determine the requirements of a speech recognition system, choosing a target domain with a reasonably limited scope that could nevertheless be regarded as a representative application domain for spoken document retrieval was a necessary first step. Considering the fact that broadcast news (BN) has been extensively used as a benchmark domain for both international speech recognition research and SDR research, Dutch television broadcast news was an obvious choice for the first explorations of Dutch SDR. In the

Focus cond.	descr.	example
F_0	Clean planned speech	television news
F_1	Clean spontaneous speech	television discussions
F_2	F_0+F_1 narrow-band	telephone interview
F_3	F_0+background music	tune in background
F_4	F_0+background noise	applause
F_5	F_0+non-native dialect	British-English
F_X	Any other combination	spontaneous non-native

Table 1.1: Focus conditions in the Hub4 "broadcast news" speech recognition evaluations.

BN domain, large variations in audio quality (microphones, bandwidth of the channels), speaker characteristics (multiple speakers, speech styles, native/non-native) and background are observed. The speech in the BN domain is often referred to as "found" speech. In the Hub4[9] benchmark tests, several *focus conditions* were distinguished, listed in Table 1.1, to enable a better error analysis. The conditions in broadcast news place great demands on the speech recognition system and typically a *large vocabulary speaker independent continuous speech recognition* (LVCSR) system is deployed for SDR tasks in this domain. In an SDR application, a speech recognition system should not produce too many errors as such a system would deliver very poor document representations that in turn will severely damage retrieval performance. At best, the document representations are exact reproductions of the words in the spoken audio, unrealistically produced by a perfect speech recognition system. The annual SDR evaluations at TREC (Text REtrieval Conference, see e.g., Garofolo et al., 2000), showed that LVCSR systems with a word error rate (WER) in the broadcast news domain between 35–40%, produced document representations that were accurate enough for retrieval. So, the envisaged Dutch speech recognition system for a Dutch multimedia retrieval system in the broadcast news domain should be capable of reaching a performance of at least 35–40% WER to enable successful retrieval.

In general, an important factor for reaching an adequate speech recognition performance for a certain language, is literally the "state-of-the-art" in speech recognition for that language that in turn is largely determined by the amount of speech (recognition) research that has been undertaken for that language. A large part of the speech recognition research has traditionally been focused on the English language which gave the development of English speech recognition systems a head start over systems for other languages. Also, the research in spoken document retrieval concentrated

[9] Hub4, with its focus on broadcast news, was an extension of the DARPA speech recognition research program based on journalistic dictation.

mainly on retrieval systems based on the English language. As the development of a speech recognition system for a specific language requires substantial investments in large corpora for system training and language specific research effort, especially for languages with fewer speakers, catching up with the performance of English systems is challenging.

For the Dutch language, a substantial amount of research has been addressed to Dutch speech (recognition) itself, both at academic and commercial sites. However, obtaining a ready-to-use, "open source" and speaker independent LVCSR speech recognition system that could be adapted to the special needs in SDR, proved to be difficult. At academic sites, the availability of a full-scale Dutch LVCSR system with a reasonable overall performance was usually not required as ASR research focused on specific parts of ASR technology, such as the acoustic modelling part (e.g., Van den Heuvel et al., 2003; De Veth, 2001) or the pronunciation generation part (e.g., Kessens, 2002; Wester, 2002), to name a few recent research topics. This type of research can well be evaluated with a minimal or partial set-up of an ASR system. Commercial Dutch speaker-independent LVCSR systems were not available either[10]. In addition, systems developed at commercial sites, usually are largely "black boxes" for commercial reasons, leaving only few or no possibilities to alter the system for research purposes. To fulfil the needs of the multimedia retrieval projects, the development of a "new" speaker independent LVCSR system for Dutch, suitable for both SDR research and LVCSR research in general, was undertaken. The purpose of this development was to deploy the system in a multimedia retrieval framework for gaining experience with Dutch speech recognition as a tool for the creation of multimedia document representations and especially, for investigating Dutch specific speech recognition issues in the context of retrieval.

1.5 This thesis

Clearly, a Dutch LVCSR system is crucial for research in Dutch spoken document retrieval and an important part of this thesis addresses the development of such a system. As retrieval performance is related to speech recognition performance, minimising the error of the overall system is an important goal. The eventual speech recognition performance depends on the sum of the performance of numerous system parts that all need to be fine-tuned and investigated separately in the context of the task domain. However, as addressing all system parts extensively was not possible within the available time, the development of the Dutch LVCSR system was restricted in two ways. Firstly, the broadcast news domain was chosen as the primary task domain, but the different speech conditions in this domain as listed in Table 1.1 on page 21, were not investigated exclusively.

[10] At least there were no indications that Dutch speaker-independent LVCSR systems had been developed for commercial exploitation. It could well be however that within R$D laboratories Dutch LVCSR research was/is investigated.

All conditions were merely piled and treated as one single “broadcast news” condition. Secondly, each system part was not be investigated in full depth as the goal was to reach a reasonable system performance within a short time. This strategy was also necessary to enable a successful contribution to the MMIR projects that served as an application framework for this research. A few issues in Dutch large vocabulary speech recognition were expected to have a relatively strong impact on both speech recognition (and retrieval) performance, including compound splitting, vocabulary selection and language modelling. As these issues had not been fully investigated in the context of Dutch LVCSR yet, they were chosen for more detailed investigation.

In spite of the limitations, the set-up of a LVCSR system for Dutch can be regarded as a useful starting point for further speech recognition research, either to investigate means to improve the “baseline” system as described in this thesis, or to investigate its application in domains other than the broadcast news domain. As broadcast news is only one interesting example of multimedia data that could be subject for retrieval, the enrolment of a LVCSR system to other application domains is an especially important prerequisite to make sure that the increasing amount of information being stored in a variety of Dutch multimedia data remains accessible.

In summary, the goals of the research described in this thesis are:

- to provide a starting point, baseline and framework for the investigation of a wide range of issues concerning spoken document retrieval, specifically for Dutch, deployed in a multimedia retrieval environment,
- to give a detailed specification of the development of a LVCSR system for Dutch, focused on the broadcast news domain, and thereby,
- to provide a baseline LVCSR system for Dutch enabling further research in this field,
- to make a contribution to Dutch LVCSR research by addressing some important issues in Dutch LVCSR: compound splitting, vocabulary selection and language modelling, and
- to demonstrate Dutch spoken document retrieval with an example SDR evaluation,

1.6 Thesis overview

In the next chapter, the concept of Spoken Document Retrieval is discussed in depth, eventually resulting in a detailed formulation of the research and development steps undertaken. Part II of this thesis describes the set-up and optimisation of the Dutch speech recognition system. Finally in Part III, different speech recognition configurations that were described in the previous part, are used in an illustrative spoken document retrieval evaluation, followed by a general discussion of the issues addressed in this thesis and a detailed overview of interesting future work both in the field of Dutch large vocabulary speech recognition and Dutch spoken document retrieval.

Chapter 2

Spoken Document Retrieval

The history of spoken document retrieval and the speech recognition techniques that have been applied in this field through the years are described in this chapter. Furthermore, this chapter discusses how the current, international, state-of-the-art in spoken document retrieval can be adopted for spoken document retrieval for Dutch. As a reference, a basic introduction to information retrieval and speech recognition is provided first.

2.1 Introduction

Research in Spoken Document Retrieval (SDR) is concerned with the representation of spoken audio in video and/or audio documents using speech recognition techniques, for application in information retrieval (IR). The goal in SDR is to gain access to the information that is "encoded" in the speech by "decoding" the speech signal to a suitable format —typically words— that can be used as a searchable representation of such documents. Section 2.5 describes how SDR research evolved from creating relatively simple document representations using a limited set of keywords, to the construction of detailed document representations using full-scale decoding of the speech signal. This evolution in SDR became possible due to the progress made in automatic speech recognition (ASR) research during the past decade. In the last section of this chapter it is argued that in order to enable SDR for Dutch, a system for Dutch ASR is required that reaches the international state-of-the-art, of which the performance level has set the agenda for the research described in this thesis.

Before addressing the evolution of SDR, a basic introduction to IR and ASR is provided first. The concepts of information retrieval are fully discussed in a vast amount of literature and extensive introductions to the subject can be found in for example Rijsbergen (1979) and Salton and McGill (1983). In Section 2.2 a brief introduction to information retrieval is provided for readers who are unfamiliar with the basic concepts of information retrieval. Section 2.3 gives a brief introduction to ASR and is not meant to be exhaustive. Only those topics that can be regarded as relevant for a good comprehension of the general problems in spoken document retrieval are introduced. For a more detailed introduction to ASR see for example Jelinek (1997) or Jurafsky and Martin (2000).

2.2 Brief introduction to IR

In this thesis, information retrieval is viewed according to the scenario depicted in Figure 2.1. It is assumed that there is a certain collection of documents, and that a user requests information about the existence and location of documents or fragments of documents in the collection that are relevant to some information need of the user. The user's request is formulated in a *query* using *natural language*—one or more words, phrases or even complete sentences—that is processed by an information retrieval system in order to suggest a number of documents that match this request. These suggestions are returned as pointers or references (e.g., hyperlinks). The retrieval system ranks these pointers according to the system's interpretation of the degree to which the suggestions match the request. The user eventually has to satisfy his information need by actually consulting the suggested items. Note that such an information retrieval system is not a so-called question-answering system (Voorhees, 2000) that attempts to provide a specific fact that the user is looking for, or to give an answer to a query. An information retrieval system as described above, typically consists of the following components:

- Document representation component
- Query formulation component
- Comparison or matching component

The document representation component deals with the conversion of the document collection to a format that is suitable for the envisaged retrieval process. From an engineering point of view, the purpose of the decoding component is to enable an efficient comparison of queries and documents in large document collections, and a real-time response to the user's query. This can be achieved by compressing the documents into a so-called *index*, which is usually a list of words (index terms) in an *inverted file* structure (Rijsbergen, 1979, page 53), where each of the indexing terms is linked to one or more documents if that term also occurs in these documents. The indexing terms may be a pre-defined set of keywords (e.g., based on a thesaurus) or simply all words in the document collections.

From an information retrieval point of view, the function of the document representation component of an IR system (depicted in Figure 2.2), is firstly to provide a document representation that can be matched against

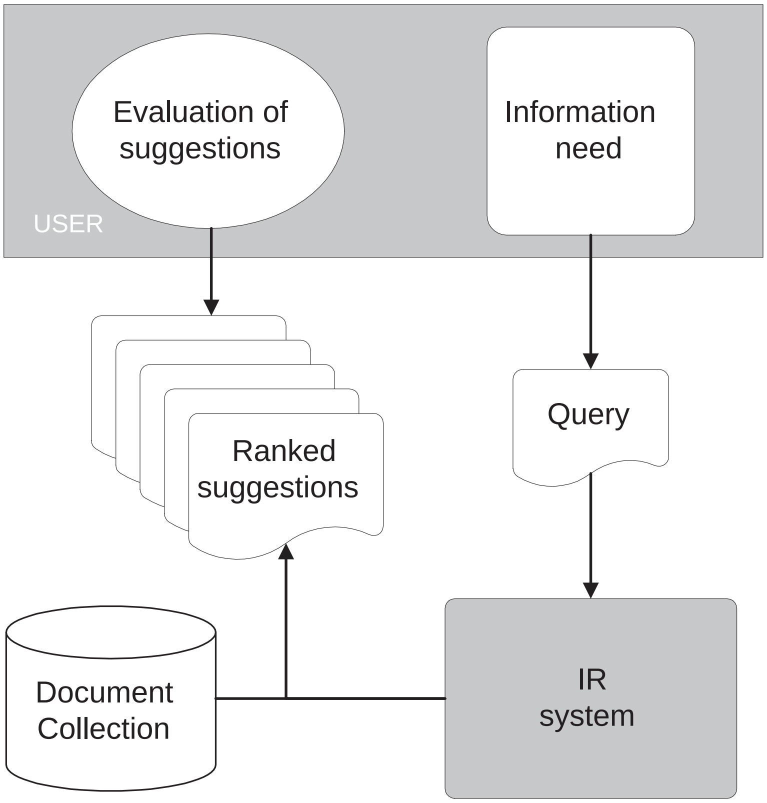

Figure 2.1: Information retrieval scenario: given a document collection, a user has some information need that is formulated in a query. This query is processed by an information retrieval system that suggests a number of documents that match the query. The user evaluates these suggestions to see whether his information need can satisfied

the representation of the query, typically a textual representation given natural language queries. A second function of this component is storing efficiently all significant information about the documents that can be used in the comparison component, including the removal of redundant information, and optionally, adding extra information (see section 2.2.2). Given a textual representation, the processes within this component include:

- tokenisation, such as the removal of punctuation marks,
- the removal of words with few document distinguishing abilities from the indexing terms, such as high frequent words and specific function words, specified in a *stop list*,
- morphological normalisation, such as stemming and compound splitting, and
- adding synonyms, such as "*Verenigde Naties* (English: United Nations)" given "*VN*".

Furthermore, search terms can be weighted in the document representation component, according to one of the many term weighting algorithms that have been developed. The weight of a term reflects the document distinguishing abilities of the term. The well-known *tf.idf* weighting scheme for example, deploys the frequency of a word in a document (term frequency *tf*) and the number of documents a term occurs in (*inverse document frequency: idf*). According to this scheme, a word that is very frequent in a certain document but also in the entire collection has only few distinguishing abilities and will receive little weight, whereas a word with a high frequency in a document and a low overall frequency, will receive more weight. These term frequencies and inverse document frequencies can also be stored in the index.

In the context of multimedia retrieval, creating a document representation that is comparable with the query representation in terms of *representation units* is a significant procedure. As described in the introductory chapter of this thesis, the representation format of a natural language query (text) does not match the representation format of images (pixels) and audio (samples). Therefore, the representation of the query and the document collection has to be put in agreement by adjusting the document representation to the query representation, by adjusting the query to the document representation or by converting both document and query representation to an intermediate representation. Note that the conversion of representations also plays a role in the context of Cross-language IR (Sheridan and Ballerini, 1996), where query and/or documents are converted to a target/intermediate language, thesaurus based IR, that maps the terms in documents and/or queries to a set of thesaurus terms (Spink and Saracevic, 1997), and cross-modal IR, which exploits the semantic relations between the various data streams in multiple media streams (Owen and Makedon, 1999; Westerveld, 2002).

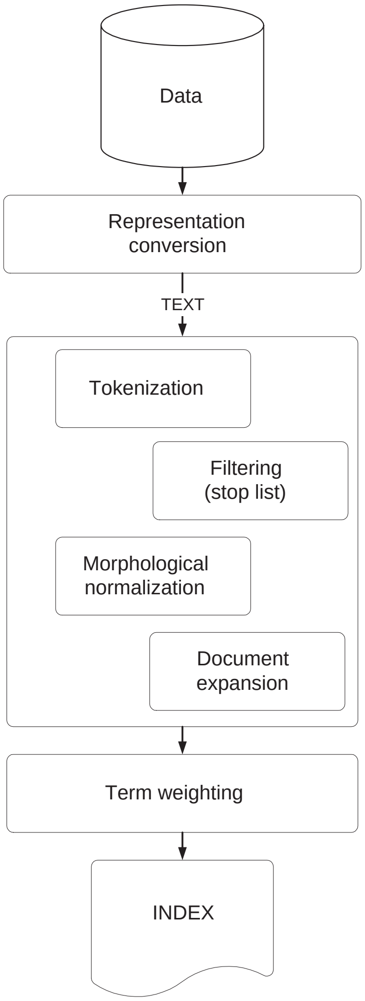

Figure 2.2: Document representation component of an IR system. If required, first the document representation is altered to prevent a representation mismatch. Given a textual representation, redundant information is removed and optionally extra information is added in a pre-processing step that precedes the actual term weighting and indexing process.

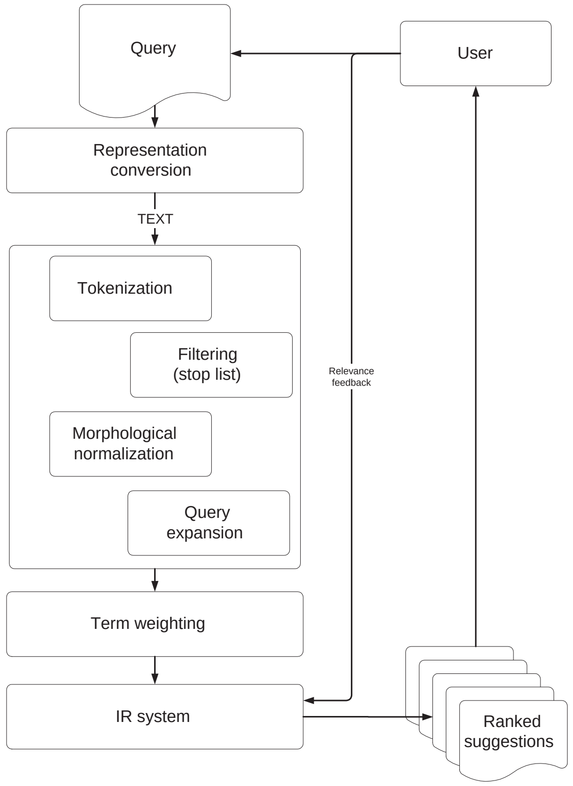

Figure 2.3: Query representation component of an IR system that broadly follows the procedures in the document representation component: representation conversion if required, pre-processing and term weighting. Optionally feedback can be provided to the user given successive query formulations

In a broad sense, the query formulation component of an IR system, depicted in Figure 2.3, involves both the creation of a query representation and performance of an *interactive dialog* with the user. The process of creating a query representation is equal to the process involved with the document representation component, including the definition of the basic representation unit, pre-processing and term weighting, based for example on the frequency of terms in the query. The dialog part, optionally includes feedback to the user given successive query formulations (referred to as *relevance feedback*).

In the comparison component of the system, a matching function compares the query representation with the document representations in the index and provides a list of documents, usually ranked according to relevance.

2.2.1 Models of IR and term weighting

The exact characteristics of the three components of an information retrieval system described above, are highly determined by the information retrieval model that is chosen. In principle, the information retrieval models provide the theoretical grounds for different approaches to the retrieval problem. The *Boolean model* was the leading model from commercial retrieval systems until the mid 1990's (Hiemstra, 2001). This model does not apply term weighting, nor does it provide a ranking of retrieved documents. In this model sets of documents are created by combining query terms with Boolean operators (AND, NOT and OR). Another example of an information retrieval model is the *vector space model* (Salton and McGill, 1983). In this model both the document representations and the query are represented as vectors in a high dimensional Euclidean space where each term is assigned a separate dimension. The similarity between a document and a query vector is typically measured by computing the cosine of the angle between both vectors. The *probabilistic* approach to information retrieval (e.g., Robertson and Spärck-Jones, 1976), stresses the importance of the ranking of suggested documents according to their probability of relevance. This probability is in principal based on the relative sizes of the subsets of documents that are indexed using the words in the query, but many probabilistic approaches have been suggested that extend the basic idea, for example by incorporating term frequency in the model. In the *language model-based* information retrieval model (Hiemstra, 2001), a unigram language model is created for each document in the collection, typically the result of interpolating a document language model with a general collection-wide language model. By computing for each document the probability that this document generates the query, a probability-ranked list of documents can be generated.

In practice, most information retrieval implementations are only in theory based on these models of information retrieval. The exact implementations are often inspired by intuitions and based on extensive studies on their behaviour in test collections. In general, term weighting is regarded as an important factor for the performance of an information retrieval system and a large number of weighting schemes, such as the *tf.idf* weighting scheme that was already mentioned earlier, have been proposed during the last 25 years. An example of a information retrieval implementation that uses *tf.idf* weighting is the popular *Okapi algorithm*[1] This algorithm is based on an extended probabilistic model theory that takes term frequency and document length into account (Robertson and Walker, 1994). Robertson and Walker experimented with a number of weighting algorithms which led to the frequently used Okapi-BM25 algorithm. The algorithm as formulated in (Robertson et al., 1998) was also used for the information retrieval experiments described in this thesis (Chapter 11 and Chapter 13).

2.2.2 Query and document expansion

A technique applied in information retrieval that is specifically worth mentioning in the context of spoken document retrieval, is *query expansion* (see e.g., Jourlin et al., 1999). As the term already suggests, this technique simply adds words to the query in order to improve retrieval performance. In query expansion the document search is basically performed twice. After an initial run, a selection of the top N most relevant documents generates a list of terms ranked by their weight (e.g., a tf.idf weight). The top T terms of this list are then added to the query and the search is repeated using the enriched query. Query expansion can be performed using retrieved documents from the same collection, or using retrieved documents from another (parallel) corpus. In the former case, query expansion is referred to as *blind relevance feedback*, in the latter it is called *parallel blind relevance feedback*. For example, as the speech recognition system in a spoken document retrieval task may have produced errors or may have missed important words, it can be useful to apply parallel blind relevance feedback using a corpus without errors—such as a manually transcribed corpus—in order to reduce retrieval misses due to speech recognition errors. In other approaches to query expansion, compound words are split (Pohlmann and Kraaij, 1996), geographic names are expanded (e.g., “The Netherlands” to “Amsterdam, . . . , Zaandam”) and hyponyms of unambiguous nouns are added (e.g., “flu, malaria, etc.” are added given “disease”) using thesauri and dictionaries (Jourlin et al., 1999). Also the opposite approach, *document expansion* is applied to alleviate the effect of speech recognition errors on retrieval performance (see e.g., Singhal and Pereira, 1999a). However, this approach does not work that well when story segmentation is unknown.

[1] The Okapi algorithm is named after the system developed at the Polytechnic of Central London in the early 1980’s, further developed at City University London and Microsoft Research.

2.2.3 Retrieval performance evaluation

The evaluation of information retrieval systems deserves special attention. The first step in the evaluation procedure involves the development of an information retrieval test collection, consisting of a set of documents, a number of queries and a definition of the query results, the list of documents the information retrieval should provide given a query, referred to as *relevance judgements*. These "answers" to the query cannot be exclusively defined and are highly subjective. For one user, a document may be a perfectly acceptable suggestion given a certain query, whereas for another, the relevance is only small. Moreover, relevance judgements for document-query pairs may change according to the provided instructions, or even the time of day. Therefore, the persons who must provide the judgements must be selected and instructed carefully in order to perform an unbiased evaluation. For the test collections that were developed for the TREC conference, a number of "judges" (among others former CIA officers) were hired to provide relevance assessments. These judges were instructed to make a binary decision on the relevance of each document given a query, and to prevent that information from other documents influenced their decisions. As for large test collections no judge can read and judge every document in the collection, only a sample of the collection is judged. For the TREC conference, the top 100 documents that were retrieved by participating systems and by judges performing manual and semi-automatic searches, referred to as the *pool*, were judged.

The performance of an information retrieval system is typically evaluated by looking per query at both *precision*, the fraction of retrieved documents that are actually relevant, and *recall*, the fraction of all relevant documents that were actually retrieved:

precision	$= \frac{r}{n}$	r: number of relevant documents retrieved
		n: number of documents retrieved
recall	$= \frac{r}{R}$	R: total number of relevant documents

The overall system performance is then determined by averaging precision and recall, over all queries. Often, precision of a system given various levels of recall is determined by fixing recall levels, for example ranging from 0 % to 100 % with 10 % intervals, resulting in a recall-precision graph that typically has a negative slope: increasing the recall results in a decrease of precision.

A retrieval evaluation using judgements as described above is costly to develop. Therefore, an alternative evaluation method can be applied, called the *known item retrieval task*, that simulates a user seeking a particular, half-remembered document in the collection. The goal is to generate a single correct document for a query rather than a set of relevant documents, which eliminates the need for expensive relevance assessments (Voorhees et al., 1997). In this evaluation method, systems are evaluated by looking at the rank at which the target documents were found.

2.3 Brief introduction to ASR

Speech recognition systems convert an acoustic signal to a sequence of words as depicted in Figure 2.4. A first step in the recognition process is usually to convert the acoustic signal to a set of spectral features (feature vectors) that capture the characteristics of the speech signal that are most important for speech recognition. In the next step, an *acoustic model*, trained on some example data (discussed in detail in Chapter 3), translates the stream of feature vectors into a stream of phones: the smallest units of speech of which words are composed. In order to find the words in this stream of phones, a *vocabulary* is needed, a dictionary that contains words that are expected to occur in the data, with their corresponding phonetic representation. Given that the conversion of a long string of phones into a set of words can be highly ambiguous—there are no word boundaries, the actual pronunciation of a word may deviate from the standard pronunciation, the recognition may have produced errors— *grammars* or *language models*, trained on example text data, may be used to disambiguate or restrict possible word candidates (discussed in detail in Chapter 6). The acoustic models and language models are trained on example data that has a close resemblance with the data for the intended task domain.

There are speech recognition systems in different flavours and many different configurations are conceivable. Some of these configurations are relatively simple, some are extremely complex. The choice of a speech recognition configuration highly depends on the characteristics of the specific task. Some of the more important parameters that characterise a task are:

Speech type

- *Isolated words versus continuous speech*. In specific tasks, users are expected to say only one word or phrase (as in telephone services) or are required to pause briefly between words (as in older dictation systems[2]). In such cases, isolated-word recognition is applied. The advantage of this type of speech recognition is that as word-boundaries are known, the search space can be narrowed down substantially. In contrast, continuous speech recognition, has a hard job finding these word-boundaries but is able to deal with fluent and dictated speech.

[2] Modern dictation systems are able to deal with continuous speech

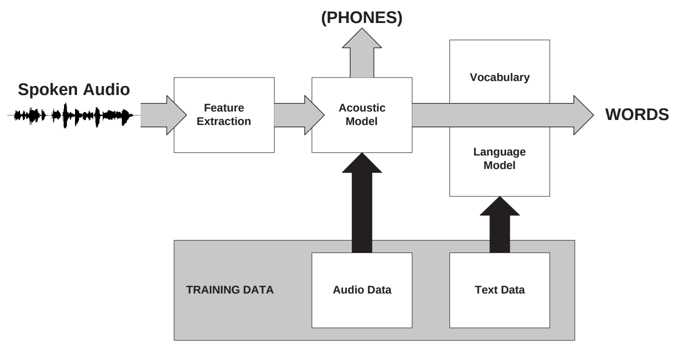

Figure 2.4: Simplified overview of a speech recognition process

- *Read speech versus spontaneous speech.* Spontaneous speech is usually less distinct and contains more disfluencies then read speech, which makes it harder to recognise correctly. Moreover, it is much more difficult to create language models that approximate spontaneous speech. Language models are usually trained on written text corpora that do not contain ungrammatical sentences and disfluencies that are frequently encountered in spontaneous speech.

Speaker dependency

- Speech is highly variable. Not only is there a large difference in speech across speakers, but within speakers as well: the speaker's physical and emotional state influences his speaking rate, pronunciation and voice quality. Across-speaker variabilities may be caused by dialect, gender or socio-linguistic background. When only one single speaker is intended to use a speech recognition system, one single acoustic model could be trained that models the speech of that speaker in all its variations. As this is virtually impossible, a general acoustic model trained on multiple speakers is often adapted to one single speaker using samples of a speaker recorded during a *speaker enrolment* session, as is common practice with most modern dictation systems. When a speech recognition system has to deal with multiple speakers, across-speaker variabilities are modelled using a large amount of data from many different speakers. In specific tasks, it can be worthwhile to train separate gender-dependent models.

Acoustic conditions

- *Bandwidth.* The frequency bandwidth of the recorded speech signal is an important factor in speech recognition as it determines how much spectral information can be used to characterise speech sounds. In narrow-band speech—typically telephone speech—only the frequency range of 300–3400 Hz is available. The absence of lower-frequency components prevents for example a proper pitch analysis, whereas accurate detection of certain phones (as fricatives) rely on the presence of the high-frequency components. Speech recognition for narrow-band speech is therefore much more difficult than for wide-band speech that also covers frequency ranges of 50–300 Hz and >3400 Hz. In tasks with multiple bandwidths, often some sort of band detection is performed (for example by computing the ratio of the average energy below and above 4kHz) so that bandwidth specific acoustic models can be applied.
- *Noise and non-speech events.* Environmental noise cannot always be banned completely. But as speech recognition systems are very sensitive to interferences of the speech signal, a lot of effort is taken either to reduce background noises as well as possible, or to filter them out of the speech signal. Furthermore, audio segments containing non-speech can introduce errors. As a speech recogniser itself cannot decide whether a particular sound is speech or just noise (or music), it may output words as a recognition result anyway. For specific applications, a speech/-non-speech detection facility can therefore be crucial for useful recognition performance.

Vocabulary requirements.

- Every task places its specific demands on vocabulary size. For some tasks a small vocabulary (up to a few hundred) is sufficient, particularly when the task can be split up into a number of subtasks, each with its own vocabulary. Other tasks require larger vocabularies, ranging from a few thousand (medium size) up to tens of thousands (large vocabulary) of words. With vocabularies becoming larger, the amount of possible sentences that can be constructed with these words grows explosively so that increasingly sophisticated grammars or language models are needed to restrict the search space. Also, word confusions are more probable as pairs of words that differ only in a few phones get more frequent in larger vocabularies.

For every speech recognition task these parameters have to be taken into account to determine a system configuration that is appropriate. The parameters that apply for retrieval tasks can put heavy demands on the configuration of a speech recognition system. Typical spoken documents such as broadcast news or voice messages contain continuous speech, often from multiple speakers who are not limited in the words they use. Also, acoustic conditions can be sub-optimal (telephone messages, live reports, music in the background). In order to deal with these difficult conditions different speech recognition approaches have been proposed. In the next sections an overview is given of these speech recognition approaches in Spoken Document Retrieval, but first two conferences that have played a mayor role in the evolution of SDR, TREC and TDT, are described in brief.

2.4 The TREC and TDT evaluations

By providing a solid infrastructure for the development and evaluation of SDR technology along with a forum for the exchange of knowledge between speech recognition and information retrieval communities (Garofolo et al., 2000), the annual NIST sponsored Text REtrieval Conference (TREC) has boosted SDR research considerably. In 1998, the TREC SDR Track was initiated as a successor to the Confusion Tracks, which aimed at the retrieval of documents whose true content has been confused or corrupted in some way (Voorhees et al., 1997). The SDR Tracks continued until 2000 (TREC-9) after which the spoken document retrieval in the broadcast news domain was declared "solved" (Garofolo et al., 2001). Although this does not mean that the spoken document retrieval problem in itself is solved—TREC focussed on one particular domain, using speech recognition for English only—it is clear that substantial progress in SDR had been made in only a few years.

Also in the DARPA Topic Detection and Tracking evaluation (Wayne, 2000) speech transcripts are a primary source. In contrast with a TREC-type retrieval system that does *retrospective retrieval* (the collection is queried *after* it is formed), TDT-type systems perform *online recognition and retrieval* in real time-longitudinal tasks, keeping track of topics, (events of interest), in a constantly expanding collection of multimedia stories. The TDT program defined five main tasks (Wayne, 2000):

- *Story Segmentation*: detect changes between topically cohesive sections

- *Topic Tracking*: keep track of stories similar to a set of example stories

- *Topic Detection*: build clusters of stories that discuss the same topic

- *First Story Detection*: detect if a story is the first story of a new, unknown topic

- *Link Detection*: detect whether or not two stories are topically linked

Because in this TDT type of tasks online recognition must be employed (recognition takes place as the audio is recorded), decoding speed is important in order to keep up with continuously incoming data streams.

2.5 ASR in spoken document retrieval

Using speech recognition technology to convert spoken audio into text for retrieval purposes, may seem a rather obvious solution. However, in order to obtain reasonable retrieval results, a speech recognition system has to produce reasonably accurate transcription of what was actually spoken. When a system produces lots of errors, successful retrieval will be doubtful. When it produces perfect transcripts, retrieval will resemble the performance of retrieving text documents. How accurate exactly speech recognition should be for acceptable retrieval performance was uncertain at the outset of SDR research, although some experience was gained with the retrieval of corrupted documents (TREC-5, e.g., Harman and Voorhees, 1997). The approach therefore was to aim at the highest performance possible. However, apart from being accurate, the speech recognition system has to perform the actual decoding within an acceptable time limit. With large data collections, decoding time grows linearly. Moreover, there is a trade off between the complexity of the recognition system and decoding speed: complex algorithms can be plugged in to improve recognition performance at the expense of processing time, or the other way around, a relatively light-weight system can save weeks of processing. It must be noted however that the rapid evolution and the decreasing costs of processor power lowered the significance of processing speed. On the other hand, because of the increasing amount of data appearing on information channels, processing speed is said to keep its relevance, especially when decoding has to be performed online, in real-time longitudinal tasks. Such tasks require that just created documents, such as today's latest broadcast news show, are searchable immediately or soon after the actual broadcast. In the TDT evaluations, this type of tasks is simulated.

Producing an acceptable performance within reasonable time limits has been outside the reach of even the most powerful speech recognition systems until very recently. Only since the mid-nineties could speech recognition be deployed increasingly successfully for spoken document retrieval tasks due to improvements in speech recognition algorithms but especially thanks to increasing amounts of computer power and memory becoming available at lower costs.

2.5.1 SDR using related text sources

Circumventing the speech recognition problem, the ORL Medusa multimedia retrieval system (Brown et al., 1995) exploited as a benchmark for purely speech-based retrieval, *teletext subtitles*. These come almost for free with a considerable amount of broadcast material. Teletext subtitles were received from the English 888 subtitles pages along with time information which, as mentioned before, is crucial to link the teletext subtitles to the video segments. The subtitles contained a nearly complete transcription of the words spoken in the material and provided an excellent information source for indexing. When available, deploying external information for the representation of spoken documents can therefore be a practical solution for a considerable amount of broadcast material. Heavy-weight speech recognition and possible retrieval performance degradation caused by speech recognition errors, can be avoided.

But in cases where subtitles or other external information sources are not available, the evident alternative is to use speech recognition technology to provide information about the documents in the collection. Different speech recognition approaches have been proposed. Each approach can be characterised by two important features: the *main representation unit* of the spoken audio in the document (document representation)–which can be words or phones for example–and the moment at which the actual decoding takes place, either *before* or *at* retrieval time. The specific advantages and disadvantages of these features are discussed in the next paragraphs that cover three main classes of speech recognition techniques in SDR: keyword spotting, sub-word based retrieval and large vocabulary speech recognition.

2.5.2 SDR using keyword spotting

Because of its relative simplicity, earliest attempts to deploy speech recognition technology in SDR made use of *word spotting* techniques to search for relevant documents in audio material (e.g., Foote et al., 1995; James, 1995; Rose et al., 1991). A keyword spotter searches the audio material for single keywords. An acoustic model is used to recognise phones and a small vocabulary of keywords with phonetic transcriptions provide the link to the keywords. Keyword searches are often weighted using a simple grammar (such as a Finite State Grammar). Weighting can be uniform for all keywords or be based on the probability distribution of the keywords in the database. Normally the spotter has a facility to reduce incorrect keyword hypotheses (*false alarms*). This may be one single “garbage” model matching all non-keywords or even a vocabulary of non-keywords.

A speech recogniser in keyword-spotter mode has the advantage of being relatively light-weight as it does not use a computationally costly language model. Therefore, keyword spotting was a feasible approach at times when computer power was still limited. In early systems, keywords were usually carefully fixed in advance. After the keyword spotting process was performed, the spoken documents in the collection could be represented in terms of the keywords found in the documents. Although, this method worked well within a very restricted domain (such as the detection of weather reports (Carey and Parris, 1995)) or topic identification in speech messages (Rose et al., 1991), the fixed set of keywords often appeared to be too limited for realistic tasks.

As computer power increased, keyword spotting could also be deployed at retrieval time, enabling the search for *any* keyword given by the user, provided that the phonetic transcriptions of the keywords could be looked up in a phonetic dictionary or successfully generated automatically by a *grapheme-to-phoneme* (G2P) converter or text-to-speech tool. However, keyword spotting at retrieval time may result in unacceptable delays in response time, especially when the document collection is large. To avoid this, James and Young (1994) proposed an alternative word spotting technique called phone lattice scanning (PLS). In PLS word spotting, phone lattices are created and searched for the sequence of phones corresponding to a particular search term. In this way keywords do not need to be chosen *a priori* so that any set of words can be searched, and as the phone lattices are created before retrieval time, delays in response time can be minimised.

But using keyword spotting for retrieval purposes has disadvantages. Retrieval will suffer from false alarms and missed keywords and especially short words are hard to spot (see e.g., Van Leeuwen et al., 1999)) as keyword spotting relies solely on acoustic information. Also, homophone occurrences as in "*In januari is de vorst ingevallen* (English: The frost came in January)" and "*In januari is de vorstin gevallen* (English: The queen fell in January)" cannot be solved without a language model or stress information. This attracted SDR researchers to use large vocabulary speech recognition systems (LVCSR, discussed below) that can benefit from the restrictive power of language models or to combine other speech recognition techniques with word spotting. Jones et al. (1996) and Brown et al. (1996) for example used LVCSR as main recognition technique and fell back to keyword spotting when out-of-vocabulary words occurred. In Ekkelenkamp et al. (1999) triphone matching (see paragraph "Sub-word unit representations" below) served as a fast but not very precise first retrieval step, after which keyword spotting was applied as a slower but more accurate retrieval refinement step.

In spite of its disadvantages, keyword spotting can be regarded as a useful technique for the retrieval of spoken documents. The focus of the SDR community however shifted toward large vocabulary speech recognition in the late nineties due to massive research efforts resulting in substantial improvements in speech recognition performance in SDR. But utilising word spotting techniques, either alone or in combination with other speech recognition techniques, remains a good choice for a variety of applications. Especially when heavy-weight speech recognition is not feasible or useful.

2.5.3 SDR using sub-word unit representations

While keyword spotting and LVCSR approaches largely focus on words as representation units of the decoded speech in the document, an alternative category of SDR approaches use sub-word unit representations such as phones, phone n-grams, syllables or broad phonetic classes (e.g., Ng, 2000; Smeaton et al., 1998) to deal with the retrieval of spoken documents. Subwords are generated by either taking the output of a phone recogniser directly (phones) or by post-processing this output to acquire phone N-grams or other representations. A significant characteristic of sub-word based approaches is that the document is represented in terms of these sub-word units. At retrieval time, query words are translated into a sequence of sub-word units which are matched with sub-word document representations.

Note that keyword spotting using a phone lattice as described earlier, resembles this type of approaches in the way that the query is translated into a sequence of sub-word units, namely phones, that are matched with the phone representation of the documents. However, keyword spotting aims at matching particular sequences of phones in the document representations themselves in order to map them to words, whereas in sub-word based approaches, the matching is done using sub-word indexing terms.

As a phone recogniser requires an acoustic model only to generate sequences of phones, the recognition process can do with a relatively simple decoding algorithm. Compared to LVCSR with computationally expensive language models, the decoding step of a sub-word based approach is therefore much faster. Also, by deploying a phone recogniser, collecting large amounts of domain specific text data (that may be unavailable) for language model training can be circumvented, which reduces training requirements to the acoustic model training. Finally, as the phone recogniser does not need a vocabulary of words, a sub-word based approach is less sensitive to out-of-vocabulary words, provided that the query words can be converted to the sub-word representations using grapheme-to-phoneme conversion tools.

However, depending solely on acoustic information, phone recognition systems tend to produce higher error rates, resulting in less accurate document representations. To compensate for the decrease in precision, hybrid approaches have been proposed, such as the one described in Ekkelenkamp et al. (1999) for example where the sub-word based approach served as a pre-selection step for a word spotting approach.

2.5.4 SDR using LVCSR

As speech recognition performance evolved to a more and more acceptable level in the late nineties, the application of large vocabulary speech recognition systems in SDR became more evident. As at the outset of the TREC SDR tracks, speech recognition performance was expected to be still relatively poor, it was questionable whether performance was good enough for reasonable retrieval performance. However, word error rates fell between 35 % and 40 % at TREC-6 which appeared to be good enough for acceptable retrieval results in a *known-item* retrieval task, simulating a user seeking one particular document. Already at TREC-7, where the known-item retrieval task was replaced by the more difficult so called *ad-hoc* retrieval task of searching multiple relevant documents from single topics, speech recognition performance was improved substantially—the University of Cambridge HTK recognition system produced error rates below the 25 % (Johnson et al., 1998)—and almost all retrieval systems performed reasonably well. Also at TREC-7, evidence could be provided for the assumption that better speech recognition performance will also result in better retrieval performance: the Cross Recogniser tasks, in which participating systems ran a retrieval experiment on speech recognition transcripts coming from different systems, showed a near-linear relationship between word error rate and retrieval performance (Garofolo et al., 2001). The same tendency was found at TREC-8.

In the TREC SDR tracks word-based systems outperformed other approaches. Out-of-vocabulary words did not appear to be a major issue: Robinson et al. (1999) reported an average OOV rate of 1 % given a 65 K word lexicon over the ad-hoc topics of TRECs 3-7. Processing time had not been a bottleneck either for the relatively heavy LVCSR systems that participated: although at TREC-8/9 the document collection was huge (557 hours of audio) and the processing (recognition and retrieval) had to be done within five months, no processing time problems were reported. Recognition error rates even dropped in comparison with the smaller TREC-7 collection. It must be noted however that there were no restrictions in the hardware or number of processors that were used. To illustrate, the HTK system of Cambridge University, the best performing system on TREC-8 with 20.5 % WER, ran in 13×RT on a Pentium III 550MHz processor running Linux (Johnson et al., 2000), which is relatively slow, especially when applied in a real time-longitudinal speech recognition task: it would take 13 days to process a single audio stream of one single day (24 hours). Therefore, in these types of tasks (or when the amount of data is simply too large in a retrospective task) speech recognition systems are optimised for speed at the cost of a certain degree of performance.

The word error rates of the speech recognition systems participating at TREC were surprisingly low given the relatively difficult broadcast news transcription task. At TREC-8, word error rates based on a 10 hour subset of the TREC-8 collection, nearly all fell between 20 % and 30 %. Apart from the increasing amount of computer power that has become available since the outset of SDR research in the mid nineties, a number of factors have contributed to these performance improvements. Firstly, TREC and other annual speech recognition related bake-off meetings[3], such as TDT and the

[3] "Bake-off" meetings were originally organised around the ARPA speech research programs. ARPA funded and other speech recognition systems were evaluated against each

ARPA Hub4 ASR task, provided an ideal framework for comparing system architectures and performances in order to reach optimal configurations for specific tasks. Moreover, as large amounts of (English) training data became available along with these evaluation meetings, research sites have, at least partly, been released from the laborious task of collecting training data for acoustic modelling and language modelling.

Next to substantial performance improvements that were achieved by refining acoustic modelling and language modelling for different speech recognition architectures, some SDR specific techniques were introduced that are especially worth mentioning:

Rolling language models: to deal with OOV words due to the daily changing focus in broadcast news, the vocabulary and language models are continuously (e.g., once a week) adapted to recent news events (Auzanne et al., 2000; Johnson et al., 2000).

Gender, bandwidth, speaker change and speech/non-speech detection: multiple detection and unsupervised adaptation techniques have successfully been devised to improve speech recognition accuracy in various ways. To improve acoustic model accuracy, gender, bandwidth and speaker change detection is performed along with the use of appropriate acoustic models for the respective conditions. Speech/non-speech detection has been applied to prevent a speech recognition system from producing (nonsense) recognition output on non-speech (e.g., Gauvain et al., 2000; Johnson et al., 2000).

Query/document expansion: when query words actually exist in a specific document, but the automatically generated transcript of the document missed these words due to speech recognition errors, this document cannot be retrieved. To compensate for such errors, query or document expansion techniques have been applied that add relevant words to the query or document to reduce the query/document mismatch. These additional relevant words are often obtained using a parallel text-based corpus, for example by running the query on the text-based corpus first, adding a selection of the words from the *N* top ranked documents of the retrieval result to the query next, and finally running the expanded query on the original document collection (see e.g., Abberley et al., 1999; Singhal and Pereira, 1999b).

Given the good performance of most SDR systems at TREC-9, the *ad hoc* retrieval task in the broadcast news domain was declared to be a “solved problem”. Summarising some general conclusions that could be noted from the TREC SDR tasks (see e.g., Garofolo et al., 2000; Johnson et al., 2000):

other for a special speech recognition task. Such competitions provide an excellent opportunity to evaluate and, if appropriate, borrow techniques that have proven to reduce error rates (Jurafsky and Martin, 2000).

- the better the performance of the speech recognition systems, the better retrieval performance in general will be[4]
- (participating) systems all perform well enough to allow standard text retrieval approaches to be successfully applied,
- word-based systems outperform systems based on sub-word units such as phones, and
- out-of-vocabulary words do not present a significant problem in the task domain.

The TREC SDR tracks have demonstrated that applying SDR techniques for the creation of multimedia/audio document representations is a valuable tool in the retrieval process of multimedia and audio documents. However, as TREC focussed on English spoken audio in only a single domain, broadcast news, there are still a number of issues that need to be addressed in SDR for other domains and other languages, as will be discussed in the next sections.

In Figure 2.5, possible SDR strategies for a given task domain and/or language that were described above are summarised, ranging from the use of speech recognition techniques (LVCSR, keyword spotting and sub-word unit based approaches) to the use of related text sources such as teletext subtitling, autocues and other types of metadata.

2.6 SDR for Dutch

In the previous section it was concluded that SDR performance in the TREC broadcast news retrieval task has grown to a more than adequate level. But although the problem might be solved for North-American broadcast news, a number of issues are still to be solved for other domains and for non-English languages, especially regarding the most important component of an SDR application, the speech recognition system. Furthermore, although the development steps that need to be undertaken to allow for SDR using LVCSR may be well-known, the actual implementation of a LVCSR system may not be straightforward, as will be shown below.

2.6.1 General issues in LVCSR development

A first, seemingly trivial requirement for SDR for whatever language or task domain, is the possession of preferably “open-source” speech recognition software that allows for adaptations given language specific or task specific characteristics. Specifically for research purposes, the adaptability of

[4] However in TREC-9 the difference in retrieval performance using query expansion, between the reference transcripts (12% WER), HTK (20% WER) and SPRACH (30%) was relatively small.

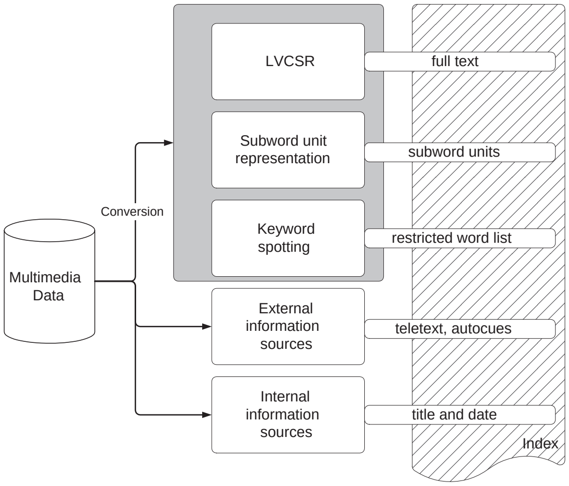

Figure 2.5: Possible SDR strategies: LVCSR, keyword spotting, phone recognition and using related text sources

an ASR system is of crucial importance. Commercial systems are usually specifically tailored for dictation tasks or telephone command and control services. Often only marginal adaptations are possible. As developing an ASR system from scratch is usually not an option, the use of existing, open-source software is the most obvious choice. As will be discussed in more detail in the next chapter, identifying such software can be difficult enough.

Given that appropriate software is available, large annotated, language and task specific, corpora are required for both acoustic model and language model training. Such corpora cannot always easily be obtained and as creating suitable collections requires substantial human effort, these are costly to develop. The huge amounts of data that were made available for English with the LVCSR bake-off meetings, are definitely not in reach for Dutch. Evidently, the number of speakers for a given language highly determines the availability of training data. For a language with fewer speakers, the commercial exploitation of corpora, and the development of systems that are created using these corpora, will be less profitable.

Next to speech recognition software and training corpora, auxiliary tools are needed for the successful deployment of a LVCSR system. To allow for the flexible construction of large speech recognition dictionaries for example, an accurate and extensive phonetic dictionary, preferably together with a grapheme-to-phoneme (G2P) processing tool for automatic word pronunciation, is indispensable. But accurate phonetic dictionaries with a large coverage are seldom freely available and costly to develop. Robust G2Ps cannot be found easily either and their development usually requires the availability of existing phonetic dictionaries. Furthermore, the large amounts of data and the relatively complex procedures that are involved in the development of a complete LVCSR system, require the use of databases and the availability of a number of specific tools, for example for scoring, language model training and annotation. Most of these tools are however open-source and can be obtained for free from a variety of sources.

Research topics focussed on obtaining the best possible speech recognition performance in the broadcast news domain. Although the TREC SDR tracks showed that successful SDR does not require a perfect speech recognition performance, it was acknowledged that targeting at a performance between 20 % and 30 % WER in the BN domain as was obtained by participating English systems in TREC, would not be entirely realistic given the available expertise, resources and man-power. Therefore, next to experiments aiming at performance improvement that follow logically the development of a LVCSR system, such as determining the optimal parameter settings in language model training, research topics were identified that could contribute to Dutch LVCSR research, instead of spending time purely on the implementation of techniques that have already proven to be successful in international ASR research.

2.6.2 Project setting

Addressing these issues has been a prerequisite for the development of a Multimedia Retrieval environment for Dutch video archives as was taken up in a series of multimedia retrieval projects: *DRUID*, *ECHO*, *MUMIS* and *Waterland* (see Appendix A for a short description). The *DRUID* project aimed at the development of tools for the indexing and retrieval of multimedia content. An important part of the project was dedicated to exploring the area of speech based retrieval for Dutch. At the outset of *DRUID* in 1998 some experience with SDR was available from the *OLIVE* project in which the speech recognition for French and German was developed by LIMSI (De Jong et al., 1999, 2000) and the take up of Dutch LVCSR recognition was an obvious next step.

Whereas in the *DRUID* project the focus is on the broadcast news domain, which in international SDR research has often served as a benchmark for system evaluations (TREC, Hub4), the focus in the *ECHO* project was on the retrospective digitisation and disclosure of *historical* national video archives. The variety of material in these archives, both acoustically and from a language modelling point of view, imposes additional requirements on both acoustic and language modelling. The audio quality of the videos recorded in the forties and fifties for example, is very poor. Also, due to the ancient vocabulary that is used in the mid-twentieth century, a substantial increase in out-of-vocabulary words and language model mismatches is generated. Finally, the different domains (documentary type of items in various domains, propaganda items, news items) and document characteristics (no strict story partitioning as in broadcast news) require fundamental adaptations to the processing scheme as opposed to the standard broadcast news approach.

The *MUMIS* project brings up other types of problems. Here the aim was to use speech recognition transcripts[5] for the indexing and retrieval of soccer matches from the Euro-2000 league. In this domain, recordings are very noisy due to stadium background noises which complicates speech recognition. Also, the limited and item-specific vocabulary (names of players, typical actions) and the a-typical type of speech (commentator's speech), require non-standard methods for vocabulary selection and language model training. Moreover, the standard LVCSR approach may not be the most appropriate solution in this domain.

Given these project setting, a speech recognition environment had to be constructed suitable for research purposes in the domain of SDR. It was decided to focus on the development of a Dutch LVCSR system, as deploying such a system as a keyword spotter or phone recogniser still remains an option, enabling the implementation of an SDR approach using keyword spotting or sub-word units respectively. It was chosen to take the English *ABBOT* system as a starting point and to port it to Dutch by coupling it with Dutch language models and acoustic models.

2.7 Research focus and thesis overview

The focus of research described in this thesis, was to set a baseline for investigating Dutch LVCSR and SDR. A major part will be devoted to the description of the necessary development steps to reach at this baseline. In addition, first explorations of this baseline as a LVCSR and SDR research framework is reported.

This thesis is structured as follows. Whereas the current part (Part I) is meant to introduce the context for the research described in the next two parts, Part II focusses on speech recognition research and development. First the *ABBOT* speech recognition system is introduced in Chapter 3. Chapter 4 describes the acquisition of a suitable phonetic dictionary and the development of a Dutch G2P. Next, acoustic model training is addressed in Chapter 5. Starting with Chapter 6 that gives an overview of n-gram modelling in speech recognition, the field of language modelling is entered. Chapter 7 gives an overview of the collected language model training data and Chapter 8 reports the development of a text normalisation module, an indispensable auxiliary tool for language model training. The selection of an optimal speech recognition vocabulary in Dutch LVCSR, is discussed in

[5] Within MUMIS, speech recognition research is conducted at the A^{2}RT group at the University of Nijmegen

Chapter 9. Chapter 10 addresses the application of compound splitting in a LVCSR framework. Applying n-gram language models in a Dutch LVCSR context is described in Chapter 11. This chapter also provides the final evaluations of the Dutch LVCSR system developed in this research. Finally, all speech recognition research and development is summarised in Chapter 12. Part III, relates the results described in Part II to Dutch SDR by providing an illustrative SDR experiment in Chapter 13. General conclusions are summarised in Chapter 14.

Part II

Speech Recognition

Chapter 3

The *ABBOT* speech recognition system

In this chapter, hybrid RNN/HMM speech recognition system ABBOT that is used in this research, will be described. First, the basic concepts of speech recognition in a probabilistic framework will be introduced. Next, the main characteristics of the ABBOT system will be outlined and compared with those of traditional HMM based systems. In the final section, the assessment methods used in this research will be addressed in brief.

3.1 Introduction

To enable spoken document retrieval research for Dutch an adaptable, open-source, Dutch speech recognition system is needed. Finding such a system however appeared to be difficult. A few commercial speech recognition systems for Dutch were available at the outset of this research, but these systems were typically tailored for speaker-dependent dictation tasks[1] or telephone command and control services[2] deploying fixed grammars and small to medium size vocabularies. Commercial large vocabulary speaker independent speech recognition systems for Dutch were not available. But even if they would have been, such systems would probably not have been very suitable for the intended research. A major drawback of commercial systems is that these are usually a "black box" for commercial reasons and cannot easily be adapted to allow for speech recognition improvement techniques that are applied frequently in SDR. Examples include language model adaptation (e.g., Auzanne et al., 2000), dynamic acoustic

[1] Speech recognition software for dictation tasks: among others, *Philips' FreeSpeech*, *Dragon's Dragon Dictate* and former *Lernout & Hauspie's VoiceExpress*

[2] Speech recognition software for telephone command and control services: among others *Philip's SpeechPearl* recognisers. Recently Nuance also released a command and control type of recogniser for Dutch

model switching (e.g., Johnson et al., 1999) or using the system as a subword based speech recognition system (e.g., Ng, 2000). Neither could the Dutch speech research community supply a system suitable for SDR purposes. Speech research in the Netherlands typically concentrates on one or several specific speech recognition topics so there is generally no need for a time-consuming set-up of a complete LVCSR system.

As a consequence of the lack of suitable Dutch systems, other options had to be explored to provide for a speech recogniser that could be used for the envisaged research. At academic sites, the open-source HMM based HTK system developed by Steve Young[3] was often chosen as a starting point for the development of a research speech recognition system, but at the outset of this research, the HTK-toolkit could not be obtained[4]. In earlier projects (Christie, 1996) some experience was gained with an English hybrid system, partly based on recurrent neural networks (RNN) and partly based on Hidden Markov Models (HMM), that by the English developers is usually referred to as the *ABBOT* system. As its sources could be obtained for research purposes it was decided to use this system and port it to Dutch. The motivation for choosing this particular system was partly based on availability. However, with the DARPA's Hub4 broadcast news speech recognition benchmark tests (Pallett, 2002) and, within an information retrieval framework, with the TREC Spoken Document Retrieval Tracks (Garofolo et al., 2000), the *ABBOT* system has also proven to be capable of reaching a top performance.

Part II of this thesis addresses the porting process of the *ABBOT* system to Dutch along with a number of language specific research topics aiming at an optimisation of the Dutch system's performance in the target domain, broadcast news. Porting a system to a target language involves the creation of language specific acoustic models and language models. As the generation of word pronunciations is a crucial procedure for both speech recognition training and decoding, word pronunciation generation is addressed first in Chapter 4. Next, the training of the Dutch acoustic models is described in Chapter 5. Starting with Chapter 6, that introduces language modelling in speech recognition, the focus is on language modelling related topics. First the collection and normalisation of language model training data is described in Chapter 7 and Chapter 8. Next, issues concerning the language model vocabulary in the broadcast news domain are closely looked at in Chapter 9 and Chapter 10. Finally, language modelling techniques and configurations are evaluated in Chapter 11, by conducting a number of broadcast news speech recognition evaluations. The results of these evaluations can be viewed at as the final Dutch speech recognition performance characteristics for the broadcast news domain that were obtained as described in the second part of this thesis.

[3] See Appendix C for a short description

[4] As all rights to HTK rested with Entropic the HTK-toolkit had been temporarily unavailable. See `http://htk.eng.cam.ac.uk/docs/history.shtml` for a brief history of HTK

In the remainder of this chapter, after a brief outline of speech recognition in a probabilistic framework, the characteristics of the hybrid RNN/HMM *ABBOT* system are described and compared with fully HMM-based systems. In the final section (Section 3.4), the speech recognition assessment methods that are used throughout this thesis are discussed in brief.

3.2 ASR in a probabilistic framework

In large vocabulary speech recognition, the task is to find the sequence of words or *sentence* W ($W = \{\omega_1, \omega_2, \ldots, \omega_N\}$) that is most likely to have been spoken on the basis of the acoustic analysis of the speech input: the acoustic observations (O). In a probabilistic framework, the probability of a sentence being produced given some acoustic observations is typically expressed as $P(W|O)$. The most probable sentence ($\hat{W}$) is found by computing $P(W|O)$ for all possible sentences and choosing the one with the highest probability:

$$\hat{W} = \arg\max P(W|O) \tag{3.1}$$

Using Bayes' rule, the conditional probability of a sentence W being spoken, assuming that certain acoustic observations O were made, can be broken down into:

$$P(W|O) = \frac{P(O|W) \cdot P(W)}{P(O)} \tag{3.2}$$

where $P(O|W)$ is the *likelihood* that specific acoustic observations are made given a sentence W, and $P(W)$ the *prior* probability of the sentence W that is obtained using *language models*. $P(O)$ is the probability of observing the given speech input. As for the computation of the most probable sentence given a certain speech input, $P(O)$ does not change, $P(O)$ may be regarded as a normalisation factor that can well be removed from the computation:

$$\hat{W} = \arg\max P(W|O) = \overbrace{P(O|W)}^{AM} \cdot \overbrace{P(W)}^{LM} \tag{3.3}$$

What marks the difference between the hybrid HMM/RNN *ABBOT* system and completely HMM-based systems, is especially the computation of $P(O|W)$, or the acoustic modelling part of the recogniser. Before the acoustic modelling itself is addressed, the input to the acoustic modelling process, the acoustic feature vectors, will be described briefly.

3.2.1 Acoustic feature extraction

To enable the computation of acoustic probabilities given some speech input, the acoustic observations first have to be defined in terms of meaningful features in a signal analysis step. First the acoustic signal is digitised by measuring its amplitude at specific intervals that is defined by its *sampling rate*, the number of samples taken per second (typically $8\,KHz$ for narrow-band, telephone speech and $16\,KHz$ for wide-band, studio speech). In a quantization process, the real-valued amplitude measurements are converted to 8-bit or 16-bit integer values. The frequency bandwidth of the recorded speech signal is an important factor in speech recognition since it determines how much spectral information can be used to characterise speech sounds. In telephone speech only the frequency range of 300–3400 Hz is available. The absence of lower-frequency components prevents for example a proper pitch analysis, whereas accurate detection of certain phones (such as fricatives) rely on the presence of the high-frequency components. Speech recognition for narrow-band speech is therefore much more difficult than for wide-band speech which also covers frequency ranges of 50–300 Hz and $>3400\,Hz$. In tasks with multiple bandwidths, some sort of band detection is often performed (for example by computing the ratio of the average energy below and above $4\,kHz$) so that bandwidth specific acoustic models can be applied. In this research the acoustic signals are digitised with a $16\,KHz$ sampling rate and stored using 16-bits integers as the majority of the speech in the task domain is recorded in a studio environment. The presence in the acoustic signal of different speech (and non-speech) sounds can best be detected using the spectral representation, the representation of the different frequency components in the signal. The acoustic observations are then typically defined in terms of a stream of *spectral feature vectors*, each representing a spectrum at a particular point in time (overlapping time-slice) using a limited set of vector coefficients. In speech recognition, smoothed versions of the actual spectrum or derivations of the spectrum, such as the cepstrum (the spectrum of a spectrum), are often used. An *LPC* spectrum (Linear Predictive Coding, Atal and Hanauer, 1971; Itakura, 1975) is an example of a smoothed spectrum. The *ABBOT* system uses feature vectors that are composed of 12 coefficients derived from an auditory-like analysis of the LPC spectrum, called PLP (Perceptual Linear Prediction, Hermansky (1990)), that modify the LPC features resembling the physic-acoustic properties of the human ear. As a 13th coefficient, a measure of the energy contained in the signal is included. The PLP features are computed from time-windows of 32 msec (512 samples), every 16 msec (256 samples).

3.2.2 Computing acoustic model probabilities

The stream of feature vectors that are extracted from the acoustic signal is passed to the next stages of the speech recognition process in order to compute the acoustic model probabilities ($P(O|W)$) and language model probabilities ($P(W)$). Using hidden Markov models to compute these probabilities is the most popular approach in speech recognition. An HMM is a stochastic automaton that consists of a set of connected states, each having a transition probability and an output or emission probability associated with it. The transition probabilities model the transitions from one state to the other. The output probabilities model the observation likelihoods of an observation being generated from a particular state. In HMM speech recognition, the problem of finding $P(O|W)$ can be expressed as finding $P(O|M)$, the likelihood that the observations O were generated by a sequence of word HMM models, M that are associated with a sentence W. The word models are in turn composed of sub-word unit models, typically *phone* models. In other words, the calculation of $P(O|W)$ involves the computation of the probability that the observations O are generated by a particular set of HMM states Q. The usual HMM training approach is to construct probability density functions (PDFs) that model the likelihood of HMM states emitting a particular observation. These PDFs are typically Gaussians or mixtures of Gaussians. The parameters of the PDFs are estimated so as to optionally model the training data. The Viterbi algorithm or alternatively a *best-first* search algorithm (stack decoding or A$*$-search), is then used to find the best path through the HMM given the observations. See for example Rabiner and Juang (1993) for a detailed survey of HMMs and search algorithms in speech recognition.

3.3 The hybrid RNN/HMM approach

The key difference between the *ABBOT* system and conventional HMM-based systems is that *connectionist*, or *neural*, networks are used to model the likelihood of HMM states emitting a particular observation, instead of probability density functions. Multi-layer perceptrons (MLP), a well-known class of neural networks, can be trained to associate an input (observation) vector with a desired output vector. Given that this output vector consists of a set of phones in a language, a trained MLP would produce a posterior probability estimate for each phone, an estimate of how probable it is that a particular input vector belongs to a phone class. These posterior probability estimates are converted to (scaled) likelihoods and then used to estimate the HMM (phone) state probabilities. In Figure 3.1, a phone probability stream given an utterance is visualised. An extensive review of the connectionist approach to speech recognition can be found in Bourland and Morgan (1994).

3.3.1 The recurrent neural net (RNN)

Instead of multi-layer perceptrons, the *ABBOT* system uses recurrent neural networks (RNN) to estimate the phone probability estimates (Robinson, 1994). The advantage of the recurrent neural net is that it can build up long term acoustic contextual information by incorporating feedback into the system (see figure 3.2). For each input frame, an acoustic vector $o(t)$ is presented as the input to the network along with the current state $q(t)$.

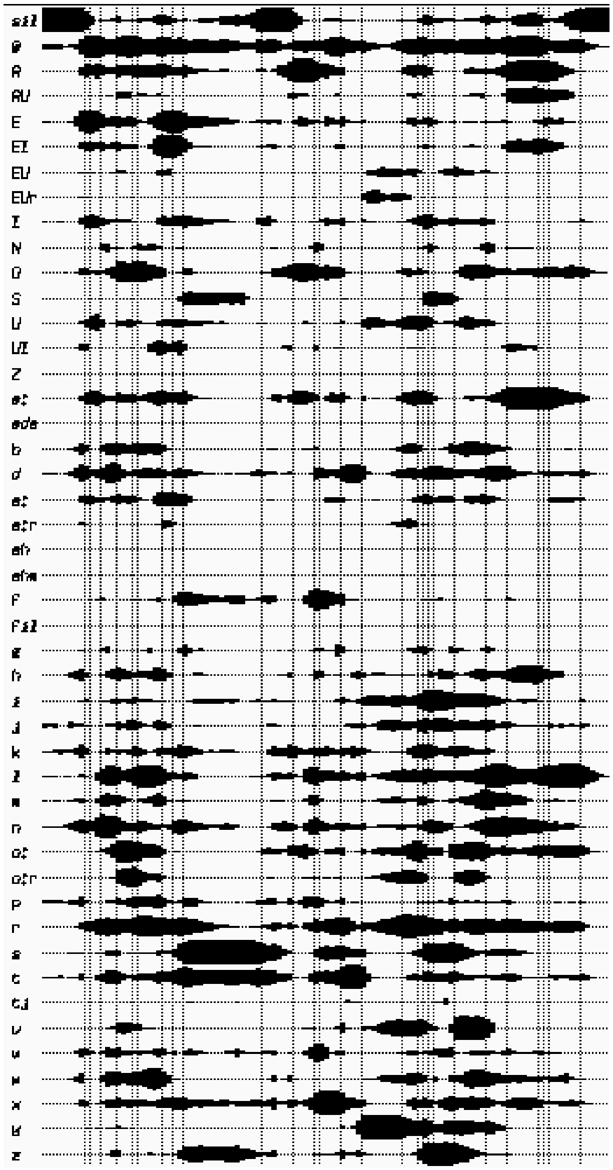

Figure 3.1: Visualisation in the log domain of the phone probability stream given the sentence “NOS Acht uur Journaal *[E n o: w E s A x t y r Z u r n a: l]*”. The figure was taken from the output of `showGuts`, an utility program that comes with the *ABBOT* software.

The network then produces an output vector $y(t)$ and the next state vector $q(t+1)$. The state vector provides the mechanism for modelling context and the dynamics of the acoustic signal as it builds up information from the start of the sequence. By delaying the output vector, for example with four frames ($y(t-4)$), forward acoustic information can be captured. State probability estimates are then obtained by estimating $P(q_k|X_1^{t+4})$, where q_k is a given state and X the observation sequence, which can be interpreted as class posterior probabilities. As Equation 3.3 requires the computation of the likelihood of the observations given a state sequence, these posterior probability estimates are converted to (scaled) likelihoods by dividing $P(q_k|X_1^{t+4})$ by the prior state probability $P(q_k)$, which can be estimated by using phone frequency information in the training data.

The phone duration is modelled within the HMM framework, where a simple Markov chain represents phone duration. As long term forward and backward acoustic context is modelled in the RNN, it is only necessary to model *context-independent* phones, resulting in a single state per context-independent phone. In HMM systems, acoustic context is modelled via *context-dependent* phones which typically results in a large number of phone models, each incorporating a number of states that model the dynamics of the signal within the phone. As a consequence a large number of parameters need to be trained.

Although using context-independent phones with the *ABBOT* system is generally sufficient, for example Cook and Robinson (1997) showed that context-dependent models may still improve performance. Context-dependent phones are chosen on the basis of a decision tree algorithm and then modelled using context-class networks (Kershaw et al., 1996). The state vector of the RNN serves as input for these context-class networks as it contains all relevant contextual information (see above). A context-dependent phone probability is estimated by computing the joint probability of the context-class probability and the phone class probability.

3.3.2 Training of the RNN/HMM system

Training of the hybrid system consists of training the RNN and the underlying Markov models. Typically a *Viterbi training* is used. This uses a forced Viterbi alignment to obtain an optimal alignment between the input frames and phone labels given the adjusted systems parameters in successive RNN training cycles. The training problem can broadly be described as finding the RNN parameters or *weights* that minimise the difference between the network outputs and the desired output. This criterion is usually represented by an error function, also called *objective function* or *criterion function*, and in a Viterbi training the error function is defined as the log posterior probability of the aligned phone sequence. By adjusting the RNN weights in the direction given by the derivative of the output error, such that the cost function is gradually reduced with respect to the RNN weights (prin-

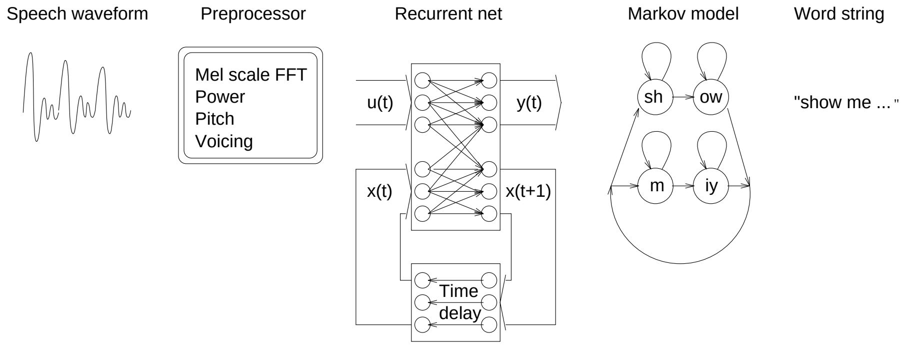

Figure 3.2: Overview of the hybrid RNN/HMM system: acoustic pre-processing, the RNN, phone duration modelling and language modelling (from Robinson et al., 1996).

ciple of gradient descent), the RNN weights are optimised. Reducing the cost function in the training process of a neural net, is done using the *back-propagation-through-time* (BPTT) algorithm (Werbos, 1990).

Although the BPTT algorithm may be very efficient in space and computation, an RNN training cycle is still computationally expensive and can best be done on dedicated hardware. Robinson et al. (2002) report that the training of a large MLP (with 8,000 hidden units) using 142 hours of training data required some 10^{15} parameter updates, and took 21 days using special-purpose hardware. It must be noted however that increasing the network size is especially useful when large amounts of training data (up to 100 hours and more) can be exploited. Given that the available Dutch training data is relatively small and no dedicated hardware could be obtained, for this research, only relatively small models were trained (restricting the RNN size to 256 feedback units, see also Chapter 5).

The transition probabilities of the phone models are optimised by reestimating the duration models and the prior phone probabilities on the pronunciations that are encountered in the training data. The training procedure is then as follows (visually depicted in Figure):

1. Assign an initial phone label to each frame of the training data by using hand-labelled speech or by *bootstrapping*. The bootstrapping method assumes that the models are already trained to some extent and are able to create a first reasonable alignment.

2. Construct the phone duration model and compute the prior phone probabilities based on the alignment.

3. Adjust the initial RNN weights so that the log posterior probability of the aligned phone sequence is maximised.

4. Apply a forced Viterbi alignment using the new parameter settings obtained in [2] and [3] and proceed with [2] until an optimum is reached.

3.3.3 Model combination

A useful feature of the hybrid framework is the possibility to combine acoustic models by merging the output of multiple networks. As the RNN is *time-asymmetric* (a standard HMM is not), training the RNN with the training data both presented forward and backward in time produces different acoustic models. It has been shown that combining the information from both models can improve performance substantially (Hochberg et al., 1994). But also other types of combinations are possible. Robinson et al. (2002) for instance, report the use of model combinations based on different acoustic features, different amounts of data and representing different balances of acoustic conditions. The combination of information sources is achieved by merging the network outputs in the log domain (Hochberg et al., 1994).

3.3.4 Word decoders: *CHRONOS* and *NOWAY*

Given a dictionary with word pronunciations combined with an acoustic model that provides the observation likelihoods $P(O|W)$, and a language model that restrains possible word sequences $P(W)$, the search for the most probable word sequence given the acoustic evidence is performed in a decoding stage. For small vocabularies and short span language models, the Viterbi algorithm can be used, but as this algorithm performs an *exhaustive* search it becomes highly inefficient when the search space expands drastically due to larger vocabularies and long span language models. Modifications to the Viterbi algorithm have been proposed, such as multi-pass (N-best or word lattice) approaches (e.g., Schwartz and Chow, 1990) to reduce the search space. These algorithms that are based on the Viterbi algorithm are called *time-synchronous*: the probabilities of active states at time t are computed *before* the probabilities at time $t + 1$ are computed. Another class of search algorithms, called best-first search algorithms, are based on *stack decoding* (or A$*$ search, Jelinek et al. (1975)). In these search algorithms, partial paths are extended in order of their probability using a search lattice or tree and keeping a priority queue (stack) of partial paths. As it is too expensive to consider all paths when applying a beam search, pruning the search is necessary. The *ABBOT* system employs confidence measures (among others by setting a threshold on the local posterior probability estimates, posterior probabilities below the threshold are pruned, Renals (1996)) for pruning and adaptive beam widths (8 in this research) are used to limit the stack size.

For the *ABBOT* system, two decoders have been developed that are based on stack-decoding, the *NOWAY* decoder that uses a *start-synchronous* search organization (see, Renals and Hochberg, 1999)) and the *CHRONOS* decoder that has implemented a *time-first* search (Robinson and Christie, 1998). As the available version of the *NOWAY* decoder[5] was substantially slower then the *CHRONOS* decoder, the latter was used for the Dutch LVCSR evaluations.

3.4 Assessment of the speech recognizer

During the porting and development procedures described in the next chapters, the (progress of the) Dutch *ABBOT* system was evaluated using a number of standard assessment techniques. As a reference, these are listed below with a short explanation.

- *Word/Phone error rate*. For the assessment of the speech recognition system as a whole the standard evaluation metric, the *word error rate* (WER02) is used. The word error rate is based upon a comparison of a *reference* transcription of the test material with the output of the recognizer referred to as the *hypothesis* transcription. The scoring algorithm searches for the *minimum edit distance* in words between the hypothesis and the reference and produces the number of substitutions, insertions and deletions that are needed to align the reference with the hypothesis. The word error rate is then defined as:

  $$WER = \frac{Insertions + Deletions + Substitutions}{Totalwordsinreference} \cdot 100 \qquad (3.4)$$

  To measure the performance of the acoustic models alone, the *phone error rate* (PER) can be computed in the same way. Instead of words, phones serve as basic units in the alignment process. The word error rates and phone error rates were obtained using the *sclite* scoring software (see Appendix C.2).

- *Term error rate*. In spoken document retrieval an alternative speech recognition error metric can be observed, the *term error rate*, that is defined as:

  $$TER = \frac{\sum_{t \in T} |R(t) - H(t)|}{|T|} \cdot 100 \qquad (3.5)$$

  where $R(t)$ and $H(t)$ represent the number of occurrences of query term t in the reference and the hypothesis respectively. The TER gives a more accurate measure of speech recognition performance conditioned on a retrieval system as it takes only the mis-recognized query terms into account.

[5] For this research, version 2.9 of the *NOWAY* decoder and version 0.5.3 (from release 1.4.0) of the *CHRONOS* decoder were available

- *Out-of-vocabulary rate*. As a measure for the quality of the speech recognition vocabulary and the language model in terms of word coverage with respect to the task, the out-of-vocabulary rate (OOV rate) is used: the ratio of the number of words in the task that do not exist in the vocabulary, to the total number of words in the task. Note that an out-of-vocabulary word that appears twice in the task, adds two counts to the amount of OOV words.

- *Perplexity*. For the evaluation of language models, the perplexity (PP) measure is used as described in detail in Section 6.5. The perplexity can be interpreted as the branching factor in the recognition task, an estimate of the number of word choices a recognizer has when it has to decide which word was spoken, or number words that are equally probable. Perplexity is thus a measure for the task difficulty from the recognizer's point of view. The less difficult the task becomes by creating language models with decreasing perplexities, the better the performance of the recognizer is expected to be (ceteris paribus).

- *Processing time*. In this research, processing time of decoding will occasionally be referred to in number of times real-time (xRT, e.g., 2.3xRT). Processing time can be an important additive performance measure, as speech recognition performance can be improved considerably when systems are not constrained in any way. However, for spoken document retrieval tasks, typically hundreds or even thousands of hours of data need to be processed so that processing time becomes very significant. In order to let a system run in real-time, it is usually necessary to restrict the system's parameters (pruning), often resulting in a considerable performance degradation. Evidently, processing time is also related to the computer system's performance. For this research, a 1GHz dual-processor Pentium-III with 1Gb RAM running Linux and a dual-processor 450 MHz Sun UltraSPARC-II workstation with 2.0 GB of memory were used.

Chapter 4

Word pronunciation generation

This chapter addresses the acquisition of word pronunciations or phonetic representations of the words in the speech recognition vocabulary. Properties of the background lexicon and the chosen phonetic representations are discussed. Finally, the development of a tool that was regarded as indispensable for this research, a Dutch grapheme-to-phoneme (G2P) converter, is described.

4.1 Introduction

Next to the acoustic model and the language model that will be addressed in the next chapters, the speech recognition dictionary is a third crucial component of an ASR system and its quality also highly determines its eventual performance. The dictionary contains the pronunciation of the words that the system can recognise. As such, the dictionary provides the link between the language model containing orthographic representations, and the acoustic models that are based on phonemic representations. Word pronunciations can be viewed as *rules* for the concatenation of phone models to arrive at the words contained in the language model (Adda-Decker and Lamel, 2000). Throughout this thesis, the list of words with pronunciations will be referred to as the speech recognition dictionary. The terms speech recognition *vocabulary* or *lexicon* will sometimes be used instead. Whereas the former explicitly includes the word pronunciations as well, the latter two terms do not.

As it is not possible to include all the words of a language in the speech recognition vocabulary, usually text data that closely resemble the task domain are deployed to obtain an indication of the word usage in the domain, enabling the selection of an appropriate set of vocabulary words. In

Chapter 9, the selection of words for the speech recognition vocabulary will be addressed in more detail. The broadcast news (BN) domain is a relatively "open" with respect to word usage. Predicting exactly which words are to be used in news items is virtually impossible and as a result of this, the usual approach is to include as many words as possible in the vocabularies, hoping that by doing so, at least the majority of the words occurring are covered. The maximum number of words that can be included in the vocabulary is restricted by the number of words a speech recognition system can deal with, which is typically 65K words. But as news topics are constantly changing, it is also necessary to revise the selection of vocabulary words with regular intervals. By doing so, words that have shown an increased news value due to recent events but were not in a vocabulary created earlier, can be recognised as well. So, for speech recognition in the BN domain, instead of using a large, static $65\,K$ vocabulary, *dynamic* vocabularies are often used that are updated frequently.

To obtain word pronunciations for the large and dynamic speech recognition vocabularies in the BN domain, speech recognition developers usually deploy a large *background* pronunciation lexicon to enable a flexible generation of word pronunciations (see Figure 4.1 on page 66). When word pronunciations are not in the background lexicon, word transcripts can be manually generated, or produced by a *grapheme-to-phoneme*[1] converter (G2P) that for instance uses rules for pronunciation generation. As generating pronunciations manually is time consuming, a G2P converter is often indispensable. Especially for languages such as Dutch or German, that have a high lexical variability due to stemming and word compounding, it is impossible to capture all possible words in a background lexicon. Obtaining word pronunciations via a G2P tool or splitting rules (see, e.g., Adda-Decker and Adda, 2000) will often be necessary. Furthermore, in the news domain, proper names and names of cities and places, referred to as *named entities*, occur frequently. But typically, only the most important named entities are included in background lexicons. Hence, for the generation of pronunciations for named entities, a G2P tool can be most helpful.

In this chapter, the generation of Dutch word pronunciations for speech recognition dictionaries deployed in the BN domain is addressed. In the next section, methods for acquiring word pronunciations are discussed in more detail including a description of the available background pronunciation lexicons. The next section shortly addresses the lexical representation of the word pronunciations that are used to develop a Dutch G2P as described in Section 4.4 and evaluated in Section 4.4.3.

[1] Also referred to as *text-to-speech* or *letter-to-phone/sound*

4.2 Acquiring word pronunciations

4.2.1 Background lexicon

Through the LDC[2] a number of pronunciation lexicons (e.g., PRONLEX for American-British English, *CELEX* for English, German and Dutch) are available for various languages that can be used as a reference for the generation of word pronunciations. Such lexicons are composed of word-pronunciation pairs. The pronunciation is provided using a particular representation or phone set (discussed below) and usually stress information (primary and secondary stress) is included[3]. The Dutch *CELEX* lexicon contains almost 120 *K* word pronunciations. However, as shown in Table 4.1, for a standard 65 *K* vocabulary, generated using the most frequent words in 300 *M* words of newspaper data (described in Chapter 7) the *CELEX* lexicon has a coverage of 37.5 %. Therefore, almost two-third of the word pronunciations have to be produced either by hand or by a G2P tool.

For the development of the grapheme-to-phoneme converter described below, the Dutch dictionary publisher *Van Dale Lexicografie*[4] provided a substantially larger pronunciation lexicon of 1,2*M* words (further referred to as *GVD*) that could be used as a background lexicon. As is shown in Table 4.1, the *GVD* lexicon has the much better coverage of 72 % given a 65 *K* word list of most frequent words in the newspaper corpus. By adding hand-crafted word pronunciations of frequent words that were missing (especially named entities and acronyms) its coverage could be improved to almost 77 % (*GVD**). Note that the pronunciation of acronyms consisting of consonants only can easily be obtained automatically by concatenating the pronunciations for every single consonant.

Reference lexicon	size	coverage	NN oov	Acron oov
CELEX	118,447	37.5 %	22.5 %	1.4 %
GVD	1,274,399	72,2 %	19.5 %	1.4 %
GVD*	1,278,361	76.6 %	17.7 %	0.3 %

Table 4.1: Coverage of background lexicons given a 65 *K* word list (top 65 *K* most frequent words in a 300 *M* words Dutch newspaper corpus) and the contribution of named entities (NN) and acronyms to the total number of missing words that are not in the lexicon.

[2] Linguistic Data Consortium, see Appendix C.1

[3] Stress information is however only sparsely used in speech recognition. A problem with modelling stress information in ASR is that it is easily confounded with higher order linguistic phenomena such as rhythmic phrasing and sentence accent (Van den Heuvel et al., 2003)

[4] See also Appendix C.1

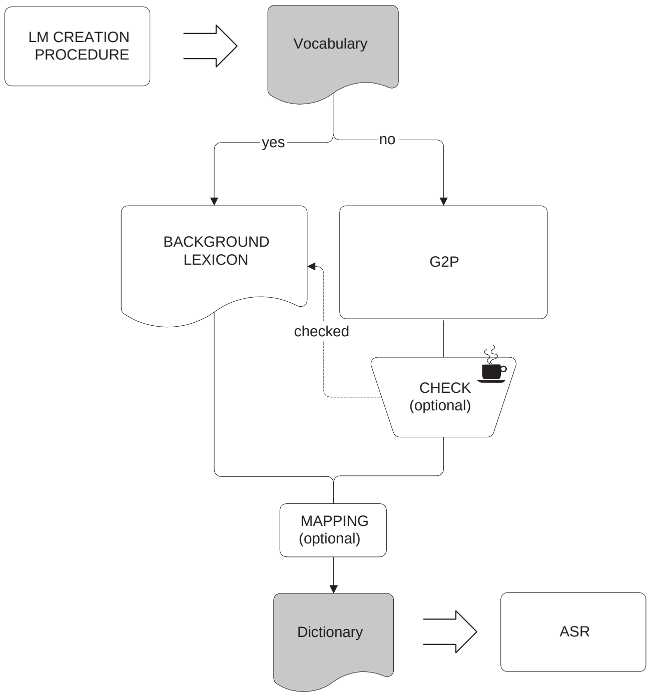

Figure 4.1: Dictionary creation procedure. The language model creation procedure typically provides vocabulary of words. The pronunciation of the words are either looked up in a background lexicon or produced by a grapheme-to-phoneme converter. In the latter case, pronunciations can be checked manually. Optionally, the pronunciation can be mapped to another phone set.

Note that a substantial part of the words that are not covered in the lexicons concern named entities and acronyms. For the *GVD** lexicon, 20 % of the missing word pronunciations can be attributed to these types of words.

4.2.2 Automatic generation of pronunciations

But even a relatively comfortable word pronunciation coverage of some 75 %, means that given a 65 K word lists well over 15 K word pronunciations are missing. As such a number of word pronunciations cannot be generated by hand, two choices remain: either including only those words in speech recognition vocabularies for which word pronunciations are available, or deploying a G2P tool that generates pronunciations automatically. As even very high performance G2P tools are bound to provide incorrect word pronunciations occasionally, relying on a G2P tool inevitably means introducing (some) pronunciation errors. On the other hand, choosing the safe side by relying on a trusted set of pronunciations provided by the background lexicon and dropping the words without pronunciations (or alternatively, manually transcribing the most frequent missing ones) will result in a decreased coverage of the words in the task domain, or a higher *out-of-vocabulary* (OOV) rate. The importance of including as many relevant words as possible in the recognition vocabulary is discussed in detail in Chapter 9. The question of whether or not to used a G2P tool, definitely depends on its performance. When the G2P tool is reasonably accurate and produces only small deviations from the correct pronunciation (such as confusing voiced with unvoiced), the speech recognition performance degradation caused by OOV words may exceed a possible degradation due to pronunciation errors.

Automatic transcription of foreign words

Foreign words—named entities such as "*New York*" or "*Moskowitch*", but also loan words—are frequently encountered in broadcast news material. A complicating factor with the automatic generation of pronunciations for these words is that they do not follow the Dutch rules for word pronunciation. G2P tools using Dutch word pronunciation rules for transcription will therefore inevitably fall short. Moreover, such words contain phonemic representations that are principally not observed in Dutch, such as the */* θ */* in "th̲e". Although the latter is mainly of concern for the definition of the units for lexical representation as discussed below, it may complicate G2P modelling based on learning algorithms as foreign phones occur relatively infrequently. Furthermore, they may introduce errors as the grapheme-to-phoneme mapping differs. For example, the grapheme sequence "th" in the Dutch word "thema (English: theme)" should be mapped to */* t */* whereas for English words this sequence is usually pronounced as */* θ */*.

4.2.3 Alternative pronunciations

Background lexicons and especially G2P tools usually provide *canonical* word pronunciations only, according to a normative, "average" pronunciation of words. However, in practice, words are pronounced in numerous variations, departing in different degradations from the canonical pronunciation, among others due to age, gender or dialect (inter-speaker variability) and speaking style, speaking rate, co-articulation or emotional state of the speaker (intra-speaker variability) (Kessens, 2002; Wester, 2002). Jost et al. (1997) estimated that in spontaneous speech around 40 % of the words are not pronounced according to the canonical representation. As such mismatches may occur both at acoustic modeling training stage and at the recognition stage, such variations result in a degradation of word accuracy of the speech recognition system (Fosler-Lussier (1999)). By incorporating pronunciation variations in the lexicon, the number of inaccurate phone-to-word mappings could be reduced.

Alternative pronunciations can be acquired using knowledge-based approaches, for instance by applying phonological rules. A Dutch example is the schwa-insertion rule that states that a schwa may be inserted in nonhomorganic consonant clusters[5] in coda position[6], such as in "melk (English: milk)" that can be observed with and without schwa-insertion: *[m ɛ l ə k]* and *[m ɛ l k]*[7]. Using this rule, an alternative pronunciation can thus be generated and added to the background lexicon. Alternatively, data-driven methods that deploy the manual transcripts of some training data or the transcripts of a phone recognition system for acquiring pronunciation variants. For an overview of the literature on pronunciation variation see Strik and Cucchiarini (1999). In Wester (2002) and Kessens (2002) rule-based and data-driven approaches applied for the Dutch language are compared.

In this research, pronunciation variation is only marginally addressed. In a few preliminary studies, the effect of modeling cross-word co-articulation in the acoustic training phase in order to improve acoustic modeling performance was investigated (Ordelman et al., 1999a,b). However, this approach was abandoned as it did not yield very promising results. Furthermore, it was decided not to incorporate pronunciation alternatives in the speech recognition vocabularies for two reasons. Firstly, as the Dutch pronunciation variation research described in Wester and Kessens had already been started at the outset of this research, it would be rather inefficient to start the same type of investigations. Secondly, as the speech recognition vocabulary for the *ABBOT* system is limited to 65536 words, adding pronunciation alternatives effectively means that fewer different words can be

[5] The place of articulation differs for each consonant in the cluster, e.g., compare *[m ɛ l k]* with *[m ɛ s t]*

[6] The *coda* is the final part, the *onset* is the start of the syllable

[7] The IPA notation will be used to represent phones throughout this thesis. Translations to other phone sets and examples can be found in Appendix B

included[8], resulting in higher OOV rates (see Section 9.2). Therefore, it was taken for granted in this research that the word pronunciation representations were not always optimal, favoring the incorporating of new words above new pronunciations.

4.3 Lexical representation

There are two approaches for representing the word pronunciations in the lexicon: using a *phonemic*, generalized representation with *phonemes* and a representation that includes *allophones*, different realizations of phonemes. The choice for one of these approaches is particularly determined by acoustic modeling decisions: is (some of the) allophonic variation explicitly modeled using separate phone models or is the representation of this variation left to a single phoneme model. The exact set of phones of phonemes (referred to as the *phone set*) is usually chosen on the basis of a standard system, such as *IPA* (International Phonetic Alphabet[9]) or *SAMPA* (Speech Assessment Methods Phonetic Alphabet[10]), which is based on IPA but made more suitable for application with computers by mapping the IPA symbols onto 7-bit printable ASCII characters. When a background lexicon or G2P tool is obtained elsewhere, it may be necessary to apply a *mapping* from the phone sets that are provided onto the phone sets that are wanted. Mapping phone sets may however result in a loss of information when phone sets differ in the amount of allophonic variation they describe.

Evidently, the language in the task domain determines the choice of the phone set to a large extent. However, the frequent occurrence of foreign pronunciations in the task domain may require that 'foreign' phones are included in the phone set, such as the *[θ]* phone that typically occurs in English words and is usually not a standard phone in a Dutch phone set. Alternatively, when a phone that would normally be in a Dutch phone set, is only exceptionally encountered (which for example is the case with the marginal vowels *[ɛː]*, *[øː]* and *[œː]*) it could well be removed from a standard phone set. The "*trainability*" of the phone is decisive in this respect. The individual phonetic realizations have to occur frequently in the available training data to enable the training of robust models. Aiming at fine-grained acoustic models with the intention of increasing acoustic model accuracy, may not be feasible as only a small number of examples can be observed in the available training data. Hence, there is a trade-off between resolution and robustness (Adda-Decker and Lamel (2000)) with respect to the choice of the phone set that is used for the actual acoustic model generation. In this research, two phone sets are referred to:

[8] The *CHRONOS* decoder that was used for this research did not allow multiple transcriptions for one single vocabulary entry. With the *NOWAY* decoder, each vocabulary item can have as many pronunciations as desired

[9] IPA homepage: `http://www.arts.gla.ac.uk/IPA/ipachart.html`

[10] SAMPA homepage: `http://www.phon.ucl.ac.uk/home/sampa/home.htm`

- the *GVD phone set*
  A large, detailed phone set that is used for the word pronunciation representations in the *GVD* lexicon.
- the *DRUID phone set*
  As the *GVD* set is too detailed and therefore not suitable for speech recognition, a smaller phone set of 45 phones, loosely based on the Dutch SAMPA phone set, was created for representing word pronunciations in the training and recognition vocabularies. In Chapter 5 the choice of the phone set will be discussed in more detail.

In Appendix B the *GVD* and DRUID phone sets are listed along with the IPA and SAMPA counterparts and including example words.

4.4 Development of the G2P converter

As discussed above, a G2P tool was regarded as an indispensable tool for the generation of word pronunciations in BN transcription tasks. Instead of using existing Dutch G2P tools (for example *TREETALK* (Busser, 1998) or *MORPA cum MORPHON* (Van Heuven and Pols, 1993)), it was decided to develop a new tool as this would enable flexible adaptations to the envisaged research and training lexicons. The Dutch dictionary publisher *Van Dale Lexicografie* provided the *GVD* lexicon for training purposes. Automatic grapheme-to-phoneme conversion has been the subject of many studies including some that specifically address G2P conversion for the Dutch language (among others, Van den Bosch, 1997; Van den Bosch and Daelemans, 1993; Bouma, 2000; Busser, 1998; Van Heuven and Pols, 1993; Stoianov and Nerbonne, 1999). Of the approaches mentioned in these studies, the one proposed by Van den Bosch was chosen as a starting point. In this approach the G2P applies a learning algorithm and uses a decision tree that is built during a *training* phase using a training lexicon. The decision tree has generalization capabilities that allow the G2P tool to provide transcriptions for words not present in the training lexicon. A number of training parameters can be used to influence the performance of the resulting decision tree.

4.4.1 Tree structure

Every node in the trained G2P decision tree has a phone associated with it that represents the best phone choice given a target grapheme in a particular graphemic context. For example, a graphemic context window of 3 left and right of the target grapheme can be used. Given the word `taxi`, a context window can be placed sequentially over each grapheme as follows:

```
  _ _ _ [t] a x i
 _ _ t [a] x i _
 _ t a [x] i _ _
 t a x [i] _ _ _
```

Here, the target grapheme is outlined in square brackets and the underscores denote fillers that are necessary near the beginning and ending graphemes of the word.

Figure 4.2 shows the G2P tree that is generated when a single word-transcription pair is used as input lexicon. The word used is the Dutch word "laat (English: late)" which is phonetically represented as *[* l a t *]*.

The '.' phone in two of the nodes in the tree is referred to as a *null phone* and acts as a *filler* phone as described below. Each node in the decision tree has a number of child nodes and the path to each of those child nodes is labeled with a grapheme. To find the corresponding phone for a grapheme in a given context, the G2P tree is traversed starting from the root of the tree using each of the graphemes in the window once. For optimization purposes the graphemes (or *features*) of a window have a certain ordering, the *feature order*. The ordering has to ensure that the most important or informative features (graphemes) are used first.

The decision tree is constructed using the IGTREE algorithm (Van den Bosch, 1997). This algorithm uses a formal notion of feature importance in classification called *information gain*. Here, 'importance' refers to the amount of information a certain feature provides. As one may expect, the target grapheme of a window and the graphemes directly surrounding the target are the most important for correct classification. The classification value of graphemes is likely to decrease with the distance from the target grapheme.

For tree construction, the IGTREE algorithm expects a feature ordering that is sorted by decreasing information gain value. The G2P tool provides a way to generate the information gain value for each of the features so that the feature order can be validated. For the Dutch language, the feature order used is: first, the target letter, then the first letter to the right of the target letter, next the first letter to the left of the target letter, next the second letter to the right of the target letter, and so on. The information gain values listed in Table 4.4.1 for each of the features support this ordering. These values were calculated from a lexicon of a little over 400,000 Dutch words. A window size of three left and right was used here.

The target grapheme is the most important feature for window classification[11]. The above ordering is a straightforward way of stressing the importance of the target grapheme and its neighboring graphemes.

[11] This is however not always the case: In Van den Bosch (1997), information gain values of window features for three other word-pronunciation sub-tasks (morphological segmentation, syllabification and stress assignment) are listed in which a letter directly left or right of the target letter is a significantly more important feature than the target itself.

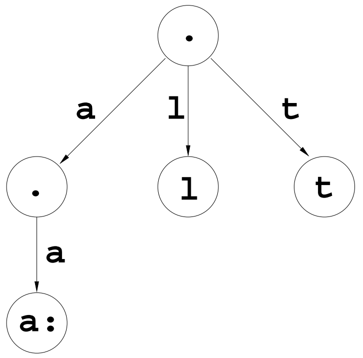

Figure 4.2: Example G2P tree

Feature	IG
T	3.500
T+1	0.869
T-1	0.687
T+2	0.307
T-2	0.198
T+3	0.116
T-3	0.060

Table 4.2: Information Gain values

4.4.2 Pre-processing of the training data

During the training/tree-construction phase, every window of a word is labeled with a phone. This requires that the number of phones in the transcription equals the number of graphemes in the word. To satisfy this requirement, some word-transcription pairs need to be pre-processed, preceding the actual training phase, applying rewrite rules and null insertion rules. Rewriting is needed for words that are shorter than their transcription. For example, the word `taxi` has four graphemes, but its Dutch transcription *[* t ᴀ k s i *]* has five phones. Therefore, using a rewrite rule that maps the grapheme `x` to `ks`, `taxi` is rewritten to `taksi`. Some forty rewrite rules are used during G2P training (see, Ordelman et al., 2001a). Another solution is needed when a word is longer than its transcription. For example, the word "*laat*" from the example tree was transcribed as *[* l a t *]*, so the transcription is one phone shorter. This can be solved by inserting a null symbol in the right place in the transcription: [l a . t]. This *null symbol* or *null phone* simply acts as a filler. About 125 null phone insertion rules were used in the G2P. The number is determined by the phone set that is used and the rewrite rules that work on the words prior to null insertion.

A disadvantage of the rewriting rules and null-phoneme insertions rules described above is that they must be learned from training errors. Errors could be found by introducing a phone mapping check. For example, a mapping of the grapheme "r" to the phone [d] is not likely for Dutch so a null-insertion error could be detected given the first example in Table 4.3. A number of trial runs are needed to obtain a useful set of rules with a low amount of errors.

r	o	n	d	d	r	i	j	v	e	n	d	e	
r	O	n	d	.	d	r	EI	v	@	n	d	@	
n	a	t	g	o	o	i	e	n					
n	A	t	x	o:	.	j	@						
c	o	m	p	j	u	t	e	r	j	u	n	k	
k	O	m	p	j	u	t	@	r	d	Z	U	N	k

Table 4.3: Training error examples

Alternative method

Currently, instead of manually creating rewrite rules and null-insertion rules, a procedure is being investigated that deploys a dynamic programming algorithm (such as minimum edit distance or a maximum probability alignment) for generating those rules automatically (e.g., Mana et al., 2001). Assigning weights or penalties to certain alignment actions (e.g., 1 for a deletion, 2 for a insertion, 1 for an allowed but non-standard grapheme-to-phone mapping and 5 for all other mappings) can then effectively be deployed to create an optimal alignment given the training data. Using training statistics, optimal penalty settings can be obtained. In this research however, the G2P implementation described above with manually generated rules is used.

4.4.3 Training and evaluation

For the training of the G2P, the *GVD* lexicon was divided into a randomly chosen training set (960 K words, ≈ 75% of the words) and a test set (320 K). Note that both sets included words with diacritics, named entities and foreign words. Stress and word boundary information was removed from the pronunciations.

In Table 4.4 the results of the G2P performance evaluation are listed. The word accuracy measure gives the percentage of pronunciations provided by the G2P that are identical to the reference pronunciations. The phone accuracy gives the percentage of phones that were correctly generated by the G2P.

discarded	testtype	#wrds	word accuracy	phone accuracy
8.06%	Train	962,170	91.2%	98.6%
-	Test	320,723	90.0%	98.5%

Table 4.4: G2P performance evaluation results including the number of discarded items during training and the number of test words.

The 90% word pronunciation accuracy of the G2P compares well with the performance of other Dutch G2P's reported in the literature. It must be noted however that the training sets and test sets used in this research are substantially larger than observed in the other studies. Although these results leave room for further improvements, the G2P was regarded as a useful and reasonably reliable tool for the generation of missing word pronunciations given the dynamic vocabularies used in the envisaged BN transcription tasks.

4.5 Summary and future work

In this chapter the generation of word pronunciations for the speech recognition dictionaries was discussed. As for broadcast news items which words are likely to be used can only be predicted to some extend, typically large vocabularies are deployed for speech recognition in this domain. As usually not all word pronunciations for the words in these vocabularies are available in a background lexicon, a grapheme-to-phoneme (G2P) converter is an indispensable tool. The development of such a G2P tool was described. The G2P applies a learning algorithm and uses a decision tree for the generation of pronunciations. The tree was constructed using the IGTREE algorithm. Although the G2P achieved a reasonable pronunciation generation accuracy of 90% for unseen words, it was explained that the training procedure, relying on a large proportion of rewriting rules and null-insertion rules, is capable of improvement. Circumventing the manual generation of the majority of these rewriting rules deploying a dynamic programming algorithm, is currently being investigated.

A G2P as dynamic speech recognition lexicon

Besides improving the training procedure of the G2P, it may be worthwhile to investigate whether pronunciation variation can be incorporated. It is well-known that pronunciation variation is a source of error in speech recognition and a number of approaches have been proposed (such as those of Kessens, 2002; Wester, 2002, for Dutch) to deal with this problem. In automatic grapheme-to-phoneme conversion, pronunciation variation is usually not addressed explicitly, as the goal is merely to enable the generation of a normative pronunciation of a word when the pronunciation cannot be derived from an, often carefully constructed, background lexicon. However, instead of regarding a G2P tool as an auxiliary tool that is only deployed in special circumstances, a G2P can also be viewed as a speech recognition lexicon itself, dynamically providing the word pronunciations that are needed at a particular stage in the recognition process. In the ideal case, the G2P provides those pronunciation variants that are most likely given some *general* knowledge that it has obtained earlier about the pronunciation of a given word, and some *task-conditioned* knowledge about the pronunciation of words, for example generated on the basis of a recognition history. The general knowledge could for example be obtained using a data-driven approach that for example deploys forced alignment techniques and decision trees for collecting pronunciation variation statistics from automatic transcriptions of a relatively large, general speech corpus, such as for Dutch the “Spoken Dutch Corpus” (CGN, see also Chapter 5). This general pronunciation knowledge could be represented using word pronunciation probabilities. The task-conditioned knowledge may then be used in two ways: firstly, to weight the probability distribution according to the local context and secondly, to adapt the general knowledge given the local pronunciation observations, resembling a continuous learning process. An example of deploying weights using task-conditioned pronunciation knowledge, could be assigning more weight to pronunciations containing certain phone deletions given that these were frequently observed in a task’s history. Although the implementation of such an approach may be complicated and undoubtedly introduces new problems (setting thresholds, incorporating phonological/phonetic knowledge), it may provide a framework for a dynamic handling of pronunciation variation in large vocabulary speech recognition tasks.

Handling loanwords and foreign names

In general, the handling of the, frequently occurring loanwords and foreign names in the BN domain, deserves some more attention in G2P development. As the pronunciation of these words often contradicts Dutch pronunciation rules, the correct word pronunciations will often not be generated by a G2P, even when these words had been included in the training set. Currently, the only solution seems to include loan words and foreign names as often as possible in a background lexicon. Preferably, one would deploy some sort of language detection and generate a pronunciation on the basis of the language classification. For example, when the language detection tool classifies a word as being an English word, an English G2P could be consulted for the generation of the pronunciation.

Compound splitting and the pronunciation lexicon

A final issue that needs to be addressed in the context of future research in grapheme-to-phoneme conversion is compound splitting. The chance that a Dutch compound word is not in the background lexicon, and hence, its pronunciation has to be obtained via a G2P tool, is generally higher than for non-compound words as new compounds can be easily invented. However, by splitting the compounds into its components, a pronunciation could still be generated from the lexicon by concatenating the available pronunciations of the components. In order to produce correct pronunciations, co-articulation rules must be applied during the concatenation process. Although compound splitting was addressed in this thesis in the context of vocabulary construction (see Chapter 10), it was not yet applied within the context of word pronunciation generation. It is worthwhile investigating how much the percentage of word pronunciations that can be provided by the background lexicon (currently some 75 %), can be improved by applying compound splitting, and a procedure for component concatenation and co-articulation correction.

Chapter 5

Acoustic model training

In this chapter the creation of Dutch acoustic models within the RNN/HMM framework is addressed. The performance of different acoustic models is discussed in relation to the training material that was used, the acoustic model merging procedure and the size of the recurrent neural net.

5.1 Introduction

The training of the acoustic models is undoubtedly one of the key procedures in the development of a speech recognition system. When it was decided to port the English *ABBOT* system to Dutch, acoustic model training was given priority and the first task became the collection of suitable acoustic training corpora. With the collected data, a number of acoustic models were trained and their performances in terms of *phone error rates* (PER) were measured on corpus specific test data and, in order to investigate which models best suit the intended task domain, broadcast news test data. As the RNN/HMM framework allows for the combination of acoustic models by merging the output of multiple networks, the effect of different model merging strategies on PER was compared. Finally, it was briefly investigated how the size of the recurrent neural net (RNN) influenced model performance. In Section 5.2, the collection of acoustic training data will be addressed, followed by a description of the research goals in the acoustic modelling part in this research. The acoustic modelling procedure using the RNN/HMM framework is outlined next, as an introduction to a review of a number of different training variants based on different corpora. Finally, the results of merging acoustic models and altering network size, are discussed.

5.2 Training data

For acoustic model training, example data is collected resembling the data characteristics in the task domain. The better the characteristics of the training data and test data match, the better speech recognition performance in general will be. But as was already discussed in brief in earlier chapters, the broadcast news (BN) domain is characterised by highly varying audio quality, speaker characteristics and acoustic conditions, as illustrated in Table 5.1. For optimal speech recognition performance in this domain, it would be preferable to train separate acoustic models for each of the observed conditions. However, this implies that these acoustic conditions can also be recognised automatically, enabling the speech recognition system to switch to the appropriate acoustic model at recognition time. As this can be complicated, only gender dependent and bandwidth dependent modelling is usually applied (e.g., Nguyen et al., 2002; Woodland, 2002) in the practise of the English Hub4 benchmark tests, without making any further distinctions in observed conditions in the broadcast news data.

Focus cond.	descr.	example
F_0	Clean planned speech	television news
F_1	Clean spontaneous speech	television discussions
F_2	F_0+F_1 narrow-band	telephone interview
F_3	F_0+background music	tune in background
F_4	F_0+background noise	applause
F_5	F_0+non-native dialect	British-English
F_X	Any other combination	spontaneous non-native

Table 5.1: Focus conditions in the Hub4 “broadcast news” speech recognition evaluations.

Still, in order to capture the acoustic variability in the broadcast news domain sufficiently, large quantities of training data are needed for acoustic model training. In the English Hub4 benchmark tests, the Linguistic Data Consortium (LDC, see also Appendix C) made hundreds of hours of BN acoustic training data available for participants (Graff, 2002). In contrast, at the start of this research, Dutch corpora suitable for speech recognition training were only marginally available. Among the ones that could be obtained were the *Groningen corpus*, a corpus of over 20 hours of read speech with speakers reading short texts and sentences containing all possible vowels and all possible consonants and consonant clusters in Dutch, and the *Speech Styles corpus*, a medium sized corpus containing spontaneous speech (monologues), semi-spontaneous speech (picture descriptions), and read speech (see Appendix D for a more detailed description of these corpora). However, as theses corpora have a small resemblance to the broadcast news domain, better matching corpora needed to be acquired. In imitation of the Hub3 DARPA research program, a speech database containing "journalistic dictation", similar to the Wall Street Journal corpus (WSJ0, Paul and Baker, 1992) was created, consisting of approximately 7 hours of read speech. This TNO-NRC speech database (see Appendix D.1) was extensively used for global tuning of the speech recognition system. To enable the training of acoustic models from BN data, the TNO-BN speech database (see Appendix D.2) was eventually also constructed, containing approximately 20 hours of speech from Dutch television broadcast news shows.

5.3 Research topics

In Section 3.3.1 it was explained that the training of the recurrent neural network (RNN) is computationally expensive and should preferably be done using dedicated hardware. An average training run for a relatively small network took days or even weeks, depending on the computer systems used. Training time could be significantly reduced by creating a multi-threaded version of the training software[1]. However, as training an RNN still took days, it was decided not to spend too much time on acoustic model training issues. Added to this, shortly after this research was initiated, in March 2000, the first release of the "Spoken Dutch Corpus" (CGN, see Appendix D.5) was published. It was expected that with this balanced corpus, acoustic model training issues could be investigated far better. However, as the preparation of such a large corpus for training is laborious, the CGN corpus was considered as a valuable training source that should be taken up in the near future, as discussed in Section 5.8.

In this context, the goals of the acoustic modelling part in this research, were defined as follows:

- acquiring "baseline" acoustic models for Dutch using the TNO-NRC speech database to enable the set-up of a basic LVCSR system for Dutch.

- acquiring broadcast news specific acoustic models using the TNO-BN speech database, that in a LVCSR set-up will be capable of producing recognition results that are good enough to address the target research questions in this thesis.

- investigating whether acoustic modelling errors can be reduced by applying model merging and altering RNN size as referred to in the literature on acoustic modelling in the hybrid RNN/HMM framework.

[1] David van Leeuwen at TNO Human Factors developed a multi-threated version of *ABBOT's* `rnnTrain`

- defining a list of appropriate research topics given the experiences in this research and the availability of the CGN corpus.

Before the results of all acoustic model training procedures are addressed, first the training procedure in a RNN/HMM framework will be outlined as a reference.

5.4 RNN/HMM training procedure

In Section 3.3.2, the training of acoustic models using the RNN architecture was already outlined. The goal of the acoustic model training was described as obtaining those RNN weights that minimise the error on a training set. Given some acoustic training data, the training procedure involves the following steps:

1. *Create feature vectors.*

   If necessary, the audio files in the training corpus are converted to the desired format (16 Khz, 16 bits in this research) and passed to the feature extraction procedure that creates for every frame of 256 samples a vector of 12 cepstral coefficients and log energy, using a window of 512 samples. The 32 bit floating point format of the PLP vectors are converted to byte coded values in a normalisation step.

2. *Label the training data and create a training set*.

   Each frame in the training data is assigned a phone label, using a forced alignment procedure. All sentences in the training data are passed to the initially untrained or partly trained RNN (in the bootstrap case), resulting in a phone probability stream per sentence. Along with a manually generated word transcription of a single sentence, the phone probability stream of a sentence is presented to a finite state grammar decoder that uses a sentence-constraint finite state grammar (FSG) to align the sentence words. After looking-up the pronunciations of the words in the sentence, a first global alignment of phones with the data is then available. The phone alignment is optimised by repeating the earlier FSG decoding step, this time however by deploying a sentence-constraint finite state grammar created using the pronunciations of the words (a phone FSG). When all phone alignments for a given training set have been made, the feature vector files created in step [1] are effectively labelled and collected in a randomised training set that can be passed to the RNN training procedure described in the next step.

3. *Start the RNN training iterations.*

   Using the labelled training data from step [2], an RNN can be trained. In this research the RNN consisted of 13 inputs (the number of PLP coefficients), 45 outputs (the number of phones in the phone set) and 256 state units (hidden layer). A single training procedure typically consists of some 30 training epochs in which the RNN parameters are gradually adjusted until an optimum is reached. After each epoch, the actual state of the network can be saved, for example to enable the generation of training statistics (phone error rate as a function of network state, as envisaged in Figure 5.1). After the completion of all training epochs, the resulting network can be passed to step [2] to generate a probability stream that is closer to the actual distribution in the training set, so that the phone alignments can be updated before a new set of RNN training epochs is started. Step [2] and [3] can then be repeated until an optimum is reached. In the training runs performed in this research, usually no substantial improvements could be observed after two iterations.

4. *Create phone models based on the training set*.

   With a (re-)labelled training set, the HMM phone models can be created or re-estimated: the *a priori* phone probabilities that are used to convert the posterior probabilities produced by the RNN to likelihoods as required by the HMM framework (see Section 3.3.1), are computed using the phone frequencies in the training set. The number of states and transition probabilities of the HMM models are estimated by measuring the mean duration of the phones in the labelled training set. After each training iteration, this procedure can be repeated in order to adjust the transition probabilities.

After training, the performance statistics of an acoustic model given some test data are obtained by performing a speech recognition run on the test data using the acoustic model and a simple phone FSG. In this FSG, every phone can be given an equal chance to be recognised, as is done in this research. Alternatively, the FSG can be based upon the phone distribution in the training data. The result of such a recognition run is a stream of most probable phones given the speech input. These phone hypotheses can then be scored[2] against phone-based reference transcripts to obtain phone error rate statistics.

5.5 Phone set

As long term forward and backward context is already modelled in the RNN, a context-independent phone set was chosen for the acoustic models with a few exceptions. The set is listed in Appendix B and is further referred to as the DRUID phone set. Some context-dependency however was introduced by including the *[* øɹ *]*, *[* eɹ *]* and *[* oɹ *]* phones, as realisations of the *[* ø *]*, *[* e *]* and *[* o *]* phones that occur before the *[* ɹ *]* and are therefore realised

[2] In this research the *sclite* scoring software was used. See also Appendix C.2

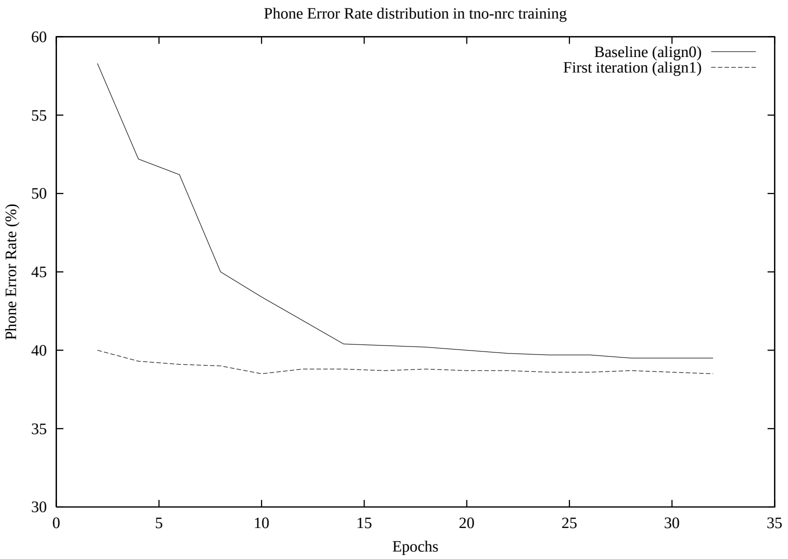

Figure 5.1: Phone Error Rate Distribution of training epochs in the *TNO-NRC* training

differently. Also, the *[* tj *]* phone were added to account for the special realisation of the diminutive suffix "-tje" as in "katje (English: little cat)". Concerning the vowels, the standard set of checked vowels and free vowels were further included. The marginal vowels *[* ɛ *]*, *[* ɛʊ *]* and *[* ɔ *]* were left out as these occur infrequently in Dutch. As for the consonants, the set included the standard set of plosives, fricatives (but without the *[* ɣ *]*, the voiced counterpart of *[x]*), and sonorants. Finally, a phone representing silence (*[sil]*) was included.

5.6 Training results

5.6.1 TNO-NRC

In the TNO-NRC training runs, 48 of the 52 speakers of the TNO-NRC speech database were used for training. The other speakers were used for testing. Instead of starting from scratch with an untrained RNN, an already trained network, based on training runs described in Christie (1996), was used for the first alignment of the training set (bootstrap weights). This was very convenient as deploying an already trained network speeds up the training procedure significantly.

In Table 5.2 the statistics of this training run are listed. In Figure 5.1 phone error rates of all training epochs in the two iterations are plotted. Both table and figure show that the phone error rate slowly improved as training progressed: from 40.1 % using the bootstrap weights to 39.5 % after a first training run of 32 epochs and finally 38.5 % after a second training run using improved alignments and phone durations.

In the last row of Table 5.2 the phone error rate (PER), and the percentages substitutions (SUB), deletions (DEL) and insertions (INS) using the TNO-NRC weights on broadcast news data was added as a reference. For this evaluation a set of 10 broadcast news shows from January–March 2002 was used. Segments containing non-speech or speech of a foreign language were excluded from the test data. In total, the test data contained approximately 3 hours of Dutch speech (See also Chapter 11). Given that the TNO-NRC consisted of read speech, it was evident that the performance of the TNO-NRC weights on broadcast news data with its large variations in speech styles would be far worse than on the TNO-NRC test data itself. However, it is striking that the number of phone substitutions increased only marginally (14 %) compared to the increase of the number of deletions (51 %). Although indeed deletions may be expected to be more frequent in spontaneous speech, the magnitude of the difference is notable.

Training	PER	SUB	DEL	INS
Bootstrap weights	40.1 %	17.0 %	19.5 %	3.6 %
After first iteration (wei-32)	39.5 %	18.2 %	17.5 %	3.9 %
After second iteration (wei-64)	38.5 %	18.4 %	16.7 %	3.4 %
Broadcast news evaluation	57.3 %	21.5 %	34.6 %	1.1 %

Table 5.2: TNO-NRC training statistics: phone error rate (PER), substitutions (SUB), deletions (DEL) and insertions (INS) in the TNO-NRC test data using the bootstrap weights and the weights resulting after a first and second training iteration. In the last row, the performance of the best TNO-NRC weights on a broadcast news evaluation set is given.

5.6.2 Groningen and Speech Styles

In Table 5.3 the evaluation results are listed for the best weights of the TNO-NRC corpus, the Speech Styles corpus and the Groningen corpus. The test data consisted of unseen data selected from the respective corpora. The differences in phone error rates in these self-tests are a result of the proportion of the within-speaker and across-speaker variation that exist in the corpus and the amount of training data that is available to model the variation. The Groningen corpus is a well balanced corpus of read speech with a large number of training examples for each phone in a particular context. The amount of data in the Speech Styles corpus is high, but the same applies for the amount of variation in the corpus, which explains the high error rate: it contains both read speech, semi-spontaneous and spontaneous speech. Note that for the broadcast news evaluation the number of deletions is again very high and the major source of recognition errors. The results confirm the expectations toward the Groningen corpus and Speech Styles corpus: these corpora do not match the data in the broadcast news domain well as the models based on this data perform worse on the BN test data than the baseline TNO-NRC models, in spite of the fact that the TNO-NRC models were created using far less training data.

Corpus	PER	SUB	DEL	INS
Groningen	26.5 %	11.0 %	11.5 %	4.1 %
TNO-NRC	38.5 %	18.4 %	16.7 %	3.4 %
Speech Styles	41.6 %	17.1 %	16.5 %	8.0 %
Groningen on BN	58.0 %	26.5 %	29.2 %	2.3 %
TNO-NRC on BN	57.3 %	21.5 %	34.6 %	1.1 %
Speech Styles on BN	59.9 %	26.6 %	31.7 %	1.5 %

Table 5.3: Evaluation statistics of the TNO-NRC, Speech Styles and Groningen training on unseen test data selected from the same corpus and on a set of 10 broadcast news shows (BN)

5.6.3 TNO-BN and Model merging

It was mentioned in Section 3.3.3 that the hybrid RNN/HMM approach allowed for the combination of acoustic models by merging the output of multiple networks. As the RNN is *time-asymmetric* (a standard HMM is not), training the RNN with the training data both presented forwards and backwards in time produces different acoustic models. For instance, Hochberg et al. (1994) showed that combining the information from both models can improve performance substantially. To investigate the effect of model merging, forwards and backwards models were trained using the TNO-NRC data and the TNO-BN data. At decoding time, the posterior probability streams of the respective forwards and backwards models were merged in the log domain. Table 5.4 shows that this procedure indeed improves acoustic model performance significantly. Evaluated on the set of 10 news broadcasts, phone error rates of the merged models are some 4–5 % lower then those of the forwards and backwards single models.

Assuming that merging models trained on different corpora and speech types could also yield some performance improvement, different merging schemes were applied that merged the probability streams of the available single and merged models. In Table 5.4 the performance on BN test data of single models and the best performing merged models are listed. It shows that training with broadcast news material is beneficial as the TNO-BN models outperform the baseline TNO-NRC models. Note that the TNO-BN models were trained using 14 hours of training data, as the full 20 hours of data became available only later (see below). The best performance is achieved by merging the forward TNO-BN model with the backward TNO-BN model. Merging the merged TNO-NRC forward and backward models with the merged TNO-BN models, in Table 5.4 referred to as TNO-NRC+BN, could not further improve performance.

Training	PER	SUB	DEL	INS
TNO-NRC (f)	57.3 %	21.5 %	34.6 %	1.1 %
TNO-NRC (b)	56.7 %	20.5 %	35.1 %	1.1 %
TNO-NRC (fb)	54.5 %	18.3 %	35.4 %	0.8 %
TNO-BN (f)	49.2 %	16.7 %	31.6 %	0.9 %
TNO-BN (b)	50.5 %	17.3 %	32.3 %	0.8 %
TNO-BN (fb)	47.3 %	14.6 %	32.1 %	0.7 %
TNO-NRC+BN (fb+fb)	49.3 w%	14.8 %	33.8 %	0.7 %

Table 5.4: Performance comparison with and without model combination. An “f” refers to a forward model, a “b” to a backward model and ’fb’ to a combined forward/backward model

5.6.4 Network size

Enlarging the size of the recurrent neural net in combination with adding more acoustic training data (such as data from the CGN corpus), can also be exploited in an attempt to improve current acoustic model performance. In Cook and Robinson (e.g., 1997); Robinson et al. (e.g., 2002) it was reported that increasing the network size can be worthwhile when large amounts of training data (up to 100 hours and more) are available. Although training data quantities up to 100 hours were definitely out of reach, the data in the TNO-BN corpus could be augmented from 14 hours up to 20 hours of broadcast news data. However, a standard acoustic model training with a network size of 256 states yielded a performance degradation relative to the same training procedure with less training data as shown in Table 5.5. Enlarging the RNN to 1024 states yielded a drastic performance improvement of almost 3 % absolute, although run time of the speech recognition system dramatically increased with it.

Training	hours	PER	SUB	DEL	INS
TNO-BN (256)	14	47.3 %	14.6 %	32.1 %	0.7 %
TNO-BN (256)	20	48.5 %	15.5 %	32.3 %	0.7 %
TNO-BN (1054)	20	44.7 %	12.8 %	31.2 %	0.6 %

Table 5.5:

5.7 General discussion and conclusions

As expected, the acoustic models trained using the corpus that matched the target domain the best, broadcast news (TNO-BN), achieved the best PER on broadcast news test data. The gain in PER compared to the best performing off-domain model, TNO-NRC, was almost 8 % absolute. The performance of the TNO-BN model, could further be improved (another 2 % absolute) by merging the forward trained TNO-BN model and the backward trained TNO-BN model. Adding more data to the broadcast news training data (an additional 6 hours) without enlarging the size of the RNN (256 units), worsened the performance of the resulting model. Only when the network size was enlarged to a hidden layer of 1024 units, did PER drop dramatically with 2.5 % absolute. However, besides the lower error rate a substantial increase in speech recognition processing time was also observed with this large RNN.

Performance of the models in this chapter was only reported in terms of PER. Although this metric was suitable for the comparison of the subsequent models, it cannot easily be related to a general notion of acoustic model “quality”. A phone error rate of almost 50 % may seem rather high, but it must firstly be recognised that this measure is obtained by defining the speech recognition output as the stream of most probable phones given some speech input. In a LVCSR setup, it is not the stream of most probable phones, but the complete phone probability stream in combination with the language model, that determines the recognition of word sequences. As will be shown in Chapter 11, a *phone* error rate of some 45–50 % broadly corresponds to some 35–40 % *word* error rate given a reasonably standard language model based on 300M words of text data. It may therefore be concluded that with the relatively small amount of training data and without spending too much time on the possible acoustic modelling related issues that are discussed in the next section, Dutch acoustic models could be trained that already allow for a very reasonable speech recognition performance. At least, performance of the models was good enough to address the targeted research questions in this thesis.

5.8 Summary and future work

In this chapter, the acoustic modelling part of the speech recognition development process using the hybrid RNN/HMM framework was addressed. It was explained that as the Spoken Dutch Corpus (CGN) was not yet available at the outset of this research, the acoustic models for this research were based on a relatively small TNO-BN broadcast news corpus especially created for this research, that nonetheless provided a reasonable performance in terms of phone error rate on the broadcast news test data. By merging the output of the RNN acoustic models trained forwards and backwards in time, performance could even be improved with some 4–5 % relative.

Increase of RNN size

By enlarging the size of the RNN from 256 to 1024 state units, model performance could be further improved, however, at the cost of a dramatic increase in speech recognition run time. Choosing a 512 hidden unit network instead of a 1024 network may reduce the increase in processing time. A 512 unit network may be expected to be sufficiently large for the amount of available training data. If the CGN corpus is to be fully deployed for acoustic model training, moving to a 1024 network may be considered again.

Exploiting the Spoken Dutch Corpus

In section 5.2, the possibility of training separate acoustic models for specific acoustic conditions, was already referred to, especially the training of gender and bandwidth dependent models, in order to improve system performance. With the relatively small amount of available training data gender and bandwidth dependent modelling was not considered for this research. However, by exploiting the CGN corpus, the training of such models comes within reach[3]. The huge amounts of data that are provided with the corpus, offer many other opportunities for acoustic modelling research. But as the speech data in the corpus is collected from a variety of domains, the question is how this data can best be exploited. One could hypothesise that deploying all available training data of the corpus for the training of acoustic models for the BN domain, will not result in a better performance than the current one obtained using a relatively small portion of domain specific training data. Data quality may be more important than data quantity with this respect. A more realistic approach could be to divide the data into a number of global domains. Domains that have a certain resemblance

[3] Note that band detection and gender detection software is required as well in order to use gender/bandwidth models in real-life (broadcast news) transcription tasks. Baseline audio segmentation software can be obtained through NIST (see Appendix C)

with the intended task domain can then be added to the training set. Alternatively, different models could be trained and merged at run-time to obtain weighted phone probability estimates based on different information sources. At least, exploiting the CGN corpus for investigating these and other issues regarding the training of Dutch acoustic models for different task domains, is promising and is regarded as one of the main topics for future research.

Chapter 6

N-gram language modelling

N-gram language models were introduced in ASR in the 1970's and still remain state-of-the-art. Also the ABBOT recogniser used in this research deploys Dutch n-gram language models for word recognition. Before the training of Dutch language models and related procedures are addressed in detail, this chapters will first provide the basic concepts of n-gram language modelling in speech recognition.

6.1 Introduction

In the previous chapters, a description was given of how the Dutch acoustic models were created and their performance was discussed in terms of phone error rates. As was pointed out in the introduction to this thesis, a language model is necessary to enable the recognition of words. In this chapter, a brief introduction to n-gram language modelling in speech recognition is presented as background information for the research described in the following chapters. For a more detailed and comprehensive introduction to language modelling see Jurafsky and Martin (e.g., 2000) or Jelinek (1976). See Chen and Goodman (1998) for an extensive comparison and detailed mathematical formulation of the described smoothing techniques.

In word recognition, the task is to find the string of words or *sentence* (W) that is most likely to have been spoken on the basis of the acoustic analysis of the speech input: the acoustic observations (O). In a probabilistic framework, the probability of a sentence being produced given some acoustic observations is typically expressed as $p(W|O)$. The most probable sentence ($\hat{W}$) is then found by computing $p(W|O)$ for all possible sentences given the vocabulary and choosing the one with the highest probability:

$$\hat{W} = \arg\max P(W|O) \tag{6.1}$$

Using Bayes' rule, the conditional probability of a sentence W being spoken, and assuming that certain acoustic observations O were made, can be broken down into:

$$P(W|O) = \frac{P(O|W) \cdot P(W)}{P(O)} \tag{6.2}$$

where $p(O|W)$ is the *likelihood* of a specific acoustic observation given a sentence W, $p(W)$ the *prior* probability of the sentence W to be produced, and $p(O)$ the probability of observing the given speech input. As for the computation of the most probable sentence given a certain speech input, $p(O)$ evidently does not change, $p(O)$ may be regarded as a normalisation factor that can well be removed from the computation. In Chapter 5 it was shown that $p(O|W)$ can be estimated using an acoustic model. In the framework of the *ABBOT* system, the recurrent neural network is deployed to generate scaled likelihoods instead of regular likelihoods. The prior probability $p(W)$ can be estimated using language models, in speech recognition, typically n-gram language models. Eventually, the task to find the sentence that is most likely to have been spoken given the acoustic analysis of the speech input ($\hat{W}$) can be formulated as:

$$\hat{W} = \arg\max P(W|O) = \overbrace{P(O|W)}^{AM} \cdot \overbrace{P(W)}^{LM} \tag{6.3}$$

6.2 *N*-gram language models

The prior probability of a sentence $p(W)$ in speech recognition is typically estimated using n-gram models. Using the chain rule of probability, $p(W)$ can be formally broken down as:

$$p(W) = \prod_{i=1}^{n} p(\omega_i|\omega_1, \ldots, \omega_{i-1}) \tag{6.4}$$

where $p(\omega_i|\omega_1, \ldots, \omega_{i-1})$ is the probability that the word ω_i was spoken, immediately *following* the preceding word sequence $\omega_1, \ldots, \omega_{i-1}$, that is referred to as the *history* of the word ω_i. Computing the probability of a word given a long history of words is however not feasible. It theoretically depends on the entire past history of a discourse. The N-gram language model attempts to provide an adequate approximation of $P(\omega_i)$ by referring to the *Markov assumption* that the probability of a future event can be predicted by looking at its immediate past. N-gram language models therefore use the previous $n - 1$ words (typically one or two words) as an approximation of the entire history. That this approximation is reasonably adequate can be derived from the fact that n-gram language models were introduced in speech recognition in the 1970's and still remain state-of-the-art. For a two-word history, *trigram* models can be generated by reformulating equation 6.4 as:

$$p(\omega) \approx p(\omega_0) \cdot p(\omega_1|\omega_0) \cdot \prod_{i=2}^{n} p(\omega_i|\omega_{i-2}, \omega_{i-1}) \tag{6.5}$$

Note that at the beginning of sentences in the training data an *start-of-sentence* mark "⟨s⟩" is usually introduced ($omega_0$ in equation 6.5) to enable the prediction of a word starting as a first word in a sentence. As no probability is assigned to ⟨s⟩ itself (it is interpreted as ω_0) it is necessary to place an *end-of-sentence* mark "⟨\s⟩" at the end of sentences (interpreted as ω_{l+1} where l is the sentence length). Otherwise, the sum of the probabilities of all strings of a given length would sum to 1 and the sum of the probabilities of all strings is then infinite. N-gram probability estimates can be computed using the relative frequencies, called *maximum likelihood estimates* (ML): the normalised counts of n-grams in a training corpus. For a trigram model that is:

$$p(\omega_3|\omega_1,\omega_2) = f(\omega_3|\omega_1,\omega_2) \doteq \frac{C(\omega_1,\omega_2,\omega_3)}{C(\omega_1,\omega_2)} \tag{6.6}$$

or in a generalised form:

$$p(\omega_i|\omega_{i-n+1}^{i-1}) = \frac{c(\omega_{i-n+1}^{i-1})}{\sum_{w_i} c(\omega_{i-n+1}^{i-1})} \tag{6.7}$$

As even very large training corpora can never cover all possible n-grams for a language, it is possible that perfectly acceptable n-grams are not encountered in the training corpus. A language model based on equation 6.6 would assign a zero probability to such n-grams. So regardless of the evidence provided by the acoustic signal in favour of an n-gram not encountered in the training data, the n-gram will be disallowed by the language model. Moreover, it is well-known that using relative frequencies as a way to estimate probabilities, produces poor estimates when the n-gram counts are small. To create a more uniform distribution, it is necessary to *smooth* these zero-probability and low-probability n-grams.

6.3 Language model smoothing

Smoothing is an important issue in language modelling and a number of smoothing algorithms are proposed in the literature. The purpose of LM smoothing is to obtain more accurate probabilities of $p(\omega)$ by adjusting the ML estimates that are obtained using the relative frequency approach. In practice this often means, reevaluating or *discounting* the n-gram counts in the corpus. Discounting refers to lowering non-zero counts according to some *discounting function* to save some *probability mass* that can be assigned to zero-counts and lower counts using a *re-distribution function*. The discounting function and the re-distribution function are typically combined using either a *back-off* strategy (Katz, 1987) or an *interpolation* strategy (Jelinek and Mercer, 1980). Both strategies use lower-order distributions for determining the probability of n-grams with zero or low counts.

6.3.1 Discounting n-gram frequency counts

An example of a simple, and generally not very well performing discounting scheme is *additive smoothing* (Jeffreys, 1948) that prevents the occurrence of zero frequencies by pretending that each n-gram occurs δ times ($0 < \delta \leq 1$) more than it does:

$$r^* = r + \delta \tag{6.8}$$

where r is the original n-gram count and r^* the discounted count. In Table 6.3.1 on page 94, bigram frequencies of frequencies (first column), counts and discounted counts are listed for a newspaper corpus of the years 1999–2001 containing some 300 M words. Given a 65 K vocabulary (V), the estimated number of bigrams that were not seen in the corpus is $V^2 - N$, where N is the number of seen bigrams. When additive smoothing is used with $\delta = 1$ (also referred to as add-one smoothing), the frequency counts are discounted as shown in the third column of the table.

A discounting scheme that is central to many other smoothing techniques is *Good-Turing* discounting (Good, 1953). In Good-Turing discounting, nonzero counts are discounted according to:

$$r^* = (r + 1)\frac{n_{r+1}}{n_r} \tag{6.9}$$

where r^* is the discounted count, n_r the number of n-grams that occurred exactly r times (frequency-of-frequency, the first column in Table 6.3.1). The Good-Turing estimate for zero frequency n-grams is than:

$$r^* = \frac{n_1}{n_0} \tag{6.10}$$

which can be converted to a probability by normalising over the original number of counts in the distribution (N):

$$p_{good\_turing} = \frac{r^*}{N} \tag{6.11}$$

Note that the Good-Turing estimate cannot be used when there are frequencies of frequencies in the distribution with a zero count. Therefore only lower counts (e.g., frequencies in the range of 0 to 7) are discounted this way, which should not be problematic as the larger counts are assumed to be reliable. The discounting range for Good-Turing discounting is normally defined by setting lower exclusion *cutoffs* for low counts (typically n-gram counts of 1, referred to as singletons) and upper discounting cutoffs. n-grams with counts above the upper discounting cutoffs are not discounted but receive the maximum likelihood estimates. In Table 6.3.1 the Good-Turing discounted counts are listed in the fourth column.

Another frequently used discounting scheme is *Witten-Bell* discounting (Witten and Bell, 1991) that estimates the probability of zero-frequency n-grams by looking at n-grams that were seen *at least* once, instead of *exactly* once as with the Good-Turing method. The idea behind this method is that zero-frequency n-grams can be modelled by the probability of seeing an n-gram for the first time. This probability can be obtained by counting the number of unique words, or word types (T), that follow a specific history. For example, the number of unique words following the word "ik (English: I)" in the Dutch newspaper corpus, is 19,347. Given a vocabulary of 65 K words, the number of unseen bigrams starting with "ik" (Z), is then 45,653. Using $\frac{N}{N+T}$, where N is the original number of n-gram counts in the distribution, as normalisation factor, counts are discounted in the following way:

$$r^* = \begin{cases} \frac{T}{Z}\frac{N}{N+T} & \text{if } r = 0 \\ r\frac{N}{N+T} & \text{if } r > 0 \end{cases} \tag{6.12}$$

In *Absolute discounting* (Ney et al., 1994) a fixed discount D ($0 \leq D \leq 1$) is subtracted from each non-zero count and re-distributed over unseen n-grams. The discount D can be estimated using the training data, but can approximated by using the estimate proposed by Ney et al., which is also used for the computation of the discounted counts in column five of Table 6.3.1:

$$D = \frac{n_1}{n_1 + 2n_2} \tag{6.13}$$

6.3.2 Model combination: interpolation and backoff

In practice, the discounting schemes discussed above are applied within a model combination framework, either using a *backoff* strategy or by *linear interpolation* of higher-order n-gram models with lower-order n-gram models. Both strategies use the notion that lower-order models can provide useful information for the computation of the probability of higher-order models, especially when there is no or insufficient data for estimating a probability for a higher-order model. With interpolation, often referred to as *Jelinek-Mercer* smoothing (Jelinek and Mercer, 1980), the unigram, bigram and trigram probabilities are mixed together, weighted by a set of weights (λ). For the estimation of a bigram probability, the interpolation formula would be:

$$p_{ip}(\omega_i|\omega_{i-1}) = \lambda p_{ML}(\omega_i|\omega_{i-1}) + (1-\lambda)p_{ML}(\omega_i) \tag{6.14}$$

As proposed by Brown et al. (1992) the linear interpolation can be defined recursively where the 1st order model (or alternatively, a uniform distribution as the 0th order model) ends the recursion:

$$p_{ip}(\omega_i|\omega_{i-n+1}^{i-1}) = \lambda_{\omega_{i-n+1}^{i-1}} p_{ML}(\omega_i|\omega_{i-n+1}^{i-1}) + (1-\lambda_{\omega_{i-n+1}^{i-1}})p_{ip}(\omega_i|\omega_{i-n+2}^{i-1}) \tag{6.15}$$

N	r	$r^{\star}_{add}$	$r^{\star}_{gt}$	$r^{\star}_{abs}$	$r^{\star}_{wb}$
4,208,024,495	0	1	0.0024	0.0028	0.4226
10,176,147	1	2	0.4599	0.315	0.9972
2,339,811	2	3	1.3537	1.315	1.9943
1,055,776	3	4	2.3290	2.315	2.9915
614,718	4	5	3.2978	3.315	3.9887
405,438	5	6	4.3192	4.315	4.9858
291,862	6	7	5.2729	5.315	5.9830
219,850	7	8	6.3468	6.315	6.9801
174,419	8	9	8	7.315	7.9773
140,753	9	10	9	8.315	8.9745
117,365	10	11	10	9.315	9.9716
99,017	11	12	11	10.315	10.9688
84,707	12	13	12	11.315	11.9660
74,143	13	14	13	12.315	12.9631
65,049	14	15	14	13.315	13.9603
57,716	15	16	15	14.315	14.9574
51,382	16	17	16	15.315	15.9546
46,001	17	18	17	16.315	16.9518
41,792	18	19	18	17.315	17.9489
38,154	19	20	19	18.315	18.9461

Table 6.1: Bigram frequencies of frequencies and discounted frequency estimates given a Dutch newspaper corpus of 300 M words, using Add-one smoothing, Good-Turing discounting, Absolute discounting and Witten-Bell discounting (conditioned on the history “ik (English: I)”. The number of unseen bigrams was estimated by subtracting the total amount of seen bigrams types from the amount of possible bigrams (V^2)

When the maximum likelihood estimates are obtained, those λ's that maximise the probability of some data can be found with the Baum-Welch algorithm (Baum, 1972) and some held-out data (held-out interpolation) or rotating parts of the training data (deleted interpolation). Note that the optimal λ values will be different for each history ω_{i-n+1}^{i-1}. As training each λ independently is not feasible, a so-called *bucketing* method can be applied, that partitions the n-grams into disjoint groups, based upon the frequency of the n-gram predicted from the lower-order models (see e.g., Bahl et al., 1989).

Katz smoothing (Katz, 1987) is regarded as the canonical example of backoff smoothing. In this smoothing method, one “backs off” to lower-order $n-1$-grams when zero-count n-grams are encountered. The key difference between interpolation and backoff is that for n-grams with *non-zero* counts, interpolation still uses information from the lower-order distributions whereas backoff does not. For bigrams, the backoff method can be represented as:

$$C_{katz}(\omega_{i-1}^{i}) = \begin{cases} r^{\star} & \text{if } r > 0 \\ \alpha(\omega_{i-1})p_{ML}(\omega_i) & \text{if } r = 0 \end{cases} \tag{6.16}$$

In order to prevent that by applying backoff the number of counts in the n-gram distribution change, the non-zero counts need to be discounted according to a discounting scheme, providing $r^{\star}$. The counts that are leftover after discounting can then be distributed among the zero-counts according to the lower-order distribution, which in the bigram case, is the unigram distribution. The value $\alpha(\omega_{i-1})$ serves as a normalisation factor (back-off weight) that must ensure that the total number of counts in the distribution remains unchanged. The α value is estimated by computing the leftover probability mass (1 minus the total probability mass) for a given n-gram context and normalising this mass by the leftover probability mass of the $n-1$-gram context as follows (from, Chen and Goodman, 1998):

$$\alpha(\omega_{i-1}) = \frac{1 - \sum_{\omega_i : c(\omega_{i-1}^{i})>0} p_{katz}(\omega_i|\omega_{i-1})}{\sum_{\omega_i : c(\omega_{i-1}^{i})=0} p_{ML}(\omega_i)} \tag{6.17}$$

which can be rewritten as:

$$\alpha(\omega_{i-1}) = \frac{1 - \sum_{\omega_i : c(\omega_{i-1}^{i})>0} p_{katz}(\omega_i|\omega_{i-1})}{1 - \sum_{\omega_i : c(\omega_{i-1}^{i})>0} p_{ML}(\omega_i)} \tag{6.18}$$

Chen and Goodman implemented the described smoothing algorithms in both backoff and interpolated versions and did an extensive comparison of the subsequent smoothing versions. One of their findings was that a modified version of Kneser-Ney smoothing consistently outperformed all other smoothing algorithms. Kneser-Ney smoothing (Kneser and Ney, 1995) is an extension of absolute discounting and tries to optimise the combination of lower-order models with higher-order models in cases where there are only a few or no counts present in the higher-order distribution. This is often illustrated in the literature using the "San Francisco" example, that could well be replaced for Dutch readers by the "Den Haag (English: The Hague)" example. "Den Haag" is a frequent bigram, the unigram "Haag" occurs almost only after the word "Den". As the unigram probability $p(Haag)$ will be high, a discounting scheme will assign a high probability to the word "Haag" after previously unseen bigram histories One can argue that this is not desirable as "Haag" is seen only in a single history. In order to optimise the unigram probability in such cases, Kneser-Ney smoothing does not use the unigram probability that is proportional to the number of occurrences of the word, but instead to the number of *different words it follows*:

$$p_{kn}(\omega_i) = \frac{N_{1+}(\bullet\omega_i)}{N_{1+}(\bullet\bullet)} \tag{6.19}$$

where $N_{1+}(\bullet\omega_i)$ is the number of different words ω_{i-1} that precede ω_i and where $N_{1+}(\bullet\bullet) = \sum_{\omega_i} N_{1+}(\bullet\omega_i)$. Chen and Goodman proposed a variation of Kneser-Ney smoothing, referred to as *modified* Kneser-Ney smoothing that uses three different parameters, D_1, D_2 and D_{3+} applied to n-grams with one, two and three or more counts respectively, instead of a single discount D for all nonzero counts. In analogy with the estimate for a single D as described with absolute discounting above, these three parameters can be obtained using the equations:

$$\begin{aligned} Y &= \frac{n_1}{n_1 + 2n_2} \\ D_1 &= 1 - 2Y\frac{n_2}{n_1} \\ D_2 &= 2 - 3Y\frac{n_3}{n_2} \\ D_{3+} &= 3 - 4Y\frac{n_4}{n_3} \end{aligned} \tag{6.20}$$

In Chapter 11, the performance of Good-Turing discounting, Witten-Bell discounting, Absolute discounting and modified Kneser-Ney smoothing is compared in a speech recognition evaluation of Dutch news shows. Modified Kneser-Ney smoothing gave the best results in terms of word error rates, although the difference was very small.

6.4 Language model adaptation

The n-gram approach is generally very successful provided there is a sufficient amount of training data available that is similar to a reasonably static task domain. When the task domain is dynamic and differs from the training conditions, achieving a good performance becomes more difficult. A number of techniques have been proposed that use *mixtures* of language models to improve language model performance under such circumstances. A typical approach is to mix language models created on a small portion of domain specific training data with language models based on a large, less domain specific, training corpus. For example, a general language model based on large amounts of newspaper data could be interpolated with a language model built from a small corpus of broadcast news transcripts for the broadcast news task domain. Interpolation weights can be fixed by estimating these using some example data, or alternatively, can be adapted dynamically as a function of the running history (e.g., Weintraub et al., 1996). Deploying the running history to improve a baseline language model is also used in the *dynamic cache model* approach, that interpolates the baseline model with a unigram cache language model based on a history of N recently observed words, in an attempt to model the phenomenon that such words have a higher probability of re-occurring (e.g., Clarkson and Robinson, 1997; Kuhn and Mori, 1990). A related approach uses *trigger pairs*, to capture information from the long-distance document history by changing the probability estimate of a trigger sequence *B* when preceded by a trigger *A* (Rosenfeld, 1996). Bellegarda (2000) proposed a hybrid language modelling approach that models word co-occurrence in *multispan language models*. Here, latent semantic analysis (LSA, see e.g., Jurafsky and Martin (2000), page 663) is embedded into the standard n-gram modelling framework.

Instead of using different corpora for mixture language models –typically one small corpus that is close to the target domain and a large one that is less specific–a single corpus, partitioned manually or automatically according to the text content, can be used. In Clarkson and Robinson (1997) for example, document clusters are created automatically using a clustering algorithm that computes the distance between a document and a cluster based upon the perplexity of a language model constructed from the cluster with respect to the document. Gotoh and Renals (2000) classify documents according to their vector representations (see also Chapter 13) using k-means clustering and in Seymore and Rosenfeld (1997) (semi-) automatic clustering is performed using topic trees. Alternatively, a corpus can be partitioned manually when document labels are available, which is usually the case with newspaper corpora.

Given a set of either manually or automatically generated document clusters, component language models can be trained. As these language models are trained on smaller text portions, the risk of data sparseness becomes more apparent: the amount of data may be too small to train robust n-gram models. In practice, there is a trade-off between data sparseness and domain similarity. When a component LM closely matches a target domain, it may outweigh the fact that it was trained using a relatively small amount of training data. Furthermore, the effect of data sparseness can be minimised by mixing component language models with a model that has been trained using the complete training corpus.

In Clarkson and Robinson (1997), the component language models are mixed by assigning weights to each of the component language models. These weights are estimated using the Expectation-Maximization (EM, see e.g., Jelinek and Mercer (1980)) on the basis of previously seen text. Among others Seymore and Rosenfeld (1997) and Gotoh and Renals (2000) apply information retrieval techniques to find the clusters that best match the topic of (a piece of) data in the task domain. Such a scheme requires the audio data being (pre-)segmented according to topic or to a specific time window. Evidently, a dual-pass decoding strategy is needed: first, an initial hypothesis transcription is generated using a baseline language model; next, this transcription is used to find the best matching document clusters so that finally an improved transcription can be generated using the topic-based mixture language model.

6.5 Evaluation of language models: perplexity

In order to measure the appropriateness of different language models for a specific speech recognition task, one could perform a set of speech recognition evaluations and determine on the basis of the respective word error rates (WER) which language model has the lowest WER and may therefore be considered as the most suitable for the task. However, as performing speech recognition evaluations is computationally expensive, such an approach is not very attractive. Moreover, word error rates are dependent on the speech recognition system which makes a comparison of language models across different speech recognition systems difficult. Therefore, language models are commonly evaluated according to their *cross-entropy* or *perplexity* on test data.

Given a language model M and a test set T, the probability of the test data given the model can be computed by taking the product of the probabilities of all sentences $(t_1, \ldots, t_T)$ in the test set:

$$p_M(T) = \prod_{i=1}^{T} p_M(t_i) \tag{6.21}$$

When comparing two language models given some test set, the language model with the highest $p_M(T)$ can be regarded as the model that best matches the test set. From information theory it is known that $-\log_2 p(T)$ bits are needed to compress a text T (Cover and Thomas, 1991). The cross-entropy H of a language model on the test set T containing W_T words, can then be defined as:

$$H_p(T) = -\frac{1}{W_T} \log_2 p(T) \tag{6.22}$$

and can be interpreted as the average number of *bits* that are needed to encode each of the words in the test data using the language model. The perplexity metric PP can be interpreted as the inverse of the average *probability* that is assigned to each word in the test set using the model and is defined as:

$$PP_p(T) = 2^{H_p(T)} \tag{6.23}$$

or in an alternative form:

$$PP_p(T) = \frac{1}{(\prod_{i=1}^{T} p(\omega_i|\omega_1 \ldots \omega_{i-1}))^{\frac{1}{T}}} \tag{6.24}$$

Using the cross-entropy or perplexity metric, language models can easily be compared without the need of going through a complete speech recognition evaluation. Given some test data that are realistic samples of the speech recognition task domain, the language model that has the lowest cross-entropy or perplexity may be expected to yield the best performance in a speech recognition task. In Chen et al. (1998) and Klakow and Peters (2002) it was shown that the correlation between word error rate and perplexity is reasonably strong.

6.6 Summary

In this chapter, a brief overview of n-gram language modelling was given as a reference for the next chapters that concern the language modelling part of the development of a Dutch speech recognition system. The basic idea of n-gram modelling in speech recognition was explained and a number of widely used smoothing techniques and model combination techniques, interpolation and backoff, were described briefly. Next, the adaptation of language models to varying task domains was addressed. A number of language model adaptation schemes were mentioned. Finally, cross-entropy and perplexity, the most widely used evaluation metrics for language models were explained.

Chapter 7

Data Collection

To enable the training of Dutch language models, large amounts of Dutch example text data are needed. This chapter describes the Dutch text data that were collected for training the language models for the broadcast news domain, including newspaper text, teletext subtitling information and the news-reader's auto-cues.

7.1 Introduction

For training robust *n*-gram statistical language models capable of modelling the great number of variations occurring in spoken language, large amounts of text data are needed, preferably data that has a close resemblance with the target domain. "There's no data like more data", is a saying in automatic speech recognition, but recently the focus on data *quantity* is slowly shifting toward data *quality* as more and more data is becoming available. Through the Linguistic Data Consortium (LDC, see Appendix C.1) for example, billions of words of (American) English news wire data and millions of words of broadcast news transcripts can be obtained (e.g., the English Gigaword Corpus, containing 12 GB of normalised English news wire text). With these amounts of training data, the advantages of acquiring even more data is expected to decrease. In language model vocabulary construction, Rosenfeld (1995) showed that better vocabularies could be constructed by focusing on data *quality*: best results were obtained when the data was carefully selected using only a fifth of the available data (see also Chapter 9). Adda et al. (1999) for instance also stress the importance of representative training data (data quality) as opposed to large amounts of training data (data quantity): texts of a 700M words newspaper database that did not lower the perplexity of the language model were eliminated.

Since the international large vocabulary speech recognition and spoken document retrieval research community directed its program largely to the BN domain (e.g., see the DARPA Hub4 benchmark tests, the TREC SDR tracks and the TDT research project), large amounts of domain specific training data have become available through the LDC: both acoustic training data and text data (newspaper data and broadcast news transcripts) for language model training (e.g., see Graff, 2002). At the outset of this research, the amounts of training data available for English were not in reach for Dutch. Existing Dutch text databases, such as the ANNO-corpus (640K words, Schuurman, 1997), the CGN corpus (10M words, Oostdijk, 2000) and INL-corpora (some 130M words, Van Dalen-Oskam et al., 2002)[1], are either too small for languages modelling purposes or not adequately available[2]. Therefore, for the training of Dutch language models, the first goal had to be improving data quantity: acquiring as much text data as possible, preferably specific to the domain of focus in this research: the broadcast news domain. The most evident source to collect was newspaper data but also other text sources were explored: teletext subtitling, auto-cues of broadcast news programs and text data from the Internet. The aim was to set-up a relatively large Dutch text database, primarily suitable for language modelling purposes, that could be made available for Dutch LM research. This database eventually became the "*Twente Nieuws Corpus*[3]" (TwNC). It contained 370M words when this thesis was completed and is still growing. Below, the text data in the corpus is described.

7.2 Newspaper data

PCM publishers[4], an organisation that administers the exploitation rights of Dutch newspapers and magazines, is providing a daily feed of newspaper articles (some 700 articles per day). Earlier, newspaper data from the years 1994-2002 was supplied from the following Dutch newspapers and magazines:

- Volkskrant
- NRC Handelsblad
- Algemeen Dagblad
- Trouw
- Parool
- Dortsch Dagblad
- Magazines: among others Elsevier and HP de Tijd

[1] See also the Bouma and Schuurman (1998) for details about existing Dutch corpora

[2] Text data of the INL for example can only be searched at INL using a search engine

[3] The Twente Nieuws Corpus is currently available through the University of Twente but will eventually be made available through the LDC.

[4] see Appendix C.1

The newspaper/magazine text data (referred to as "newspaper data") was formatted in SGML. Due to editorial differences this format was not consistent over all titles and it needed to be converted to a uniform format (in XML). In the SGML format additional information about the individual articles was included. For example, an article could have a so called "human transcription" with an article classification (e.g., "Science; Recreation; Traveling" or "International Law; International Armed Conflicts"), geographic names and organisations mentioned in the article and in which section (e.g., "Sports" or "Economics") the article was published in the newspaper. This information was preserved with the conversion to the XML format, since it can be used for a controlled selection of text data (e.g., to train language models for different domains).

7.3 Broadcast news transcripts

The collection of newspaper data provided a good starting point for the creation of language models. However, to obtain training data more similar to the broadcast news domain, other text sources were explored. Obviously, accurate broadcast news transcripts would be best suited for LM training but these are very costly to develop. *Teletext subtitling* of broadcast news programs however, can relatively easily be collected[5] and resembles the spoken text in broadcast news shows (or other television items that are subtitled), although in a somewhat abbreviated form (see below), reasonably well. Since 1998 teletext subtitling information has been collected of Dutch broadcast news programs (*NOS Acht Uur Journaal*, *Jeugdjournaal*) and news related programs (*2Vandaag*, *NOVA*). The teletext subtitling information was captured as raw text. To structure the subtitling information it was converted to a standard XML format containing labels for a date, a news-reader (if known), separate stories and sentences. Both stories and sentences also have a start-time and an end-time (parallel to the time of broadcast). Another Dutch text source that could be collected were *auto-cues* from broadcast news programs provided by the Dutch National Broadcast Foundation ([6]). In theory, auto-cues should provide a perfect match of what is actually said by the news-reader, but news-readers sometimes deviate from the cues to a greater or lesser degree. Moreover, the texts of reporters on location are normally not provided. Nevertheless, the auto-cues were regarded as a welcome addition to the training data set. The auto-cues were received in PDF format and converted to a standard XML format similar to the teletext material.

To obtain more insight into the usability of teletext subtitling and auto-cues for language modelling, it was examined how closely the teletext subtitling and auto-cues match a manual transcript of broadcast news pro-

[5] Teletext subtitling was collected using a QQS teletext capturing card (`http://www.qqs.nl`)

[6] see Appendix C.1

grams. Taking the three different sources side by side, a number of things could be observed:

- Due to a minimum of available space for subtitles on a screen, the number of words in *teletext subtitling* transcripts are cut down drastically. Phrases are often mixed up completely in an attempt to say the same with less and often other words. Teletext subtitling does cover speech in live commentaries.
- Obviously, the *auto-cues* do usually not contain transcripts of speech in live commentaries, so a number of lines will be missing in comparison with manual transcripts of a news program. However, those that are *not* missing seem to match the manual transcript (what was actually said) reasonably well.

To compare the resemblance of auto-cues and manual transcript, both versions of a single broadcast news program were taken. First, an alignment was done by hand on the phrase level to create pairs of phrases. Next, a word alignment was performed using dynamic programming (DP) to score these pairs of phrases. Empty phrases in the auto-cue transcript (possibly a text of a reporter on location) were excluded from the scoring computation.

In Table 7.1 the results are shown for the alignment procedure. Only 11% of the words in the auto-cues do not correspond with the manual transcript and half of the phrases were exactly the same. The news-reader in this news program apparently stuck closely to his or her cues. A closer look at the auto-cues transcript explains why most of the mismatches (almost 5%) are due to deletions: auto-cues are often a little short-handed. For example, the actual spoken phrase “Dames en heren, ik wens u nog een prettige avond (English: Ladies and gentleman, have a nice evening)” is abbreviated as “Goedenavond (English: Good evening)”. As far as it was possible to draw a conclusion from a comparison of only one news program and also keeping in mind that results are news-reader dependent, auto-cues transcripts seem to be fairly close to manual transcripts, as far as the news-reader’s text is concerned.

Total words	1679	
Insertions	39	2.3%
Substitutions	67	4.0%
Deletions	82	4.9%
Total errors	188	11.2%
Total sents.	124	
Sent. errors	62	50.0%

Table 7.1: Alignment score of manual transcript and auto-cues transcript of one broadcast news program

As in the teletext transcript the phrases are mixed up and other words are used, performing a word alignment is not very useful. When going through the transcripts manually, the resemblance with manual transcripts appears to be high with respect to content. See for example the following sentences, one taken from the manual transcript and one from the corresponding teletext transcript ([DEL] in the teletext sentence means that the corresponding word in the manual sentence should be "deleted" when a word alignment had been performed).

- er werden in de loop van het onderzoek drie mensen gearresteerd maar die bleken niets met de zaak te maken te hebben (English: Three people were arrested during the investigation, but it became apparent that they had nothing to do with the case) - manual transcript
- er werden [DEL] [DEL] [DEL] [DEL] [DEL] [DEL] drie mensen opgepakt maar die hadden niets met de zaak te maken [DEL] [DEL] (English: Three people were caught but they had nothing to do with the case) - teletext subtitling

Given the high similarity between manual transcripts and auto-cue transcripts, the latter can be a useful substitute for manual transcripts when these are unavailable. The texts *not* spoken by the news reader are however left aside. Teletext transcripts are far less suitable for n-gram training since phrases are shortened and mixed up with regard to the original version. On the other hand, teletext transcripts can be regarded as a useful global representation of the content of a news program. Moreover, since time information is included in teletext transcripts, these can be useful for substituting speech recognition transcripts in a spoken document retrieval task.

7.4 Internet

For some time, text data was automatically downloaded daily from some Dutch newspaper sites[7] on the Internet. It was however decided to stop this procedure: firstly, because the Internet data largely overlaps the newspaper data that was already received; secondly, because the structure of these Internet pages changed frequently, as a result of which the automatic download process had to be checked regularly.

7.5 Summary and conclusions

For the training of Dutch language models, about 370M words of text data were collected from various sources:

[7] e.g., Volkskrant: http://www.volkskrant.nl; NRC Handelsblad: http://www.nrc.nl; ANP: http://www.anp.nl

- Newspaper data provided by PCM publishers (360*M* words)
- Teletext subtitling using a teletext capturing cart (6.6*M* words)
- Auto-cues form the NOS (1.2*M* words)

The collection is reasonably large and may be regarded as an approximation of the collections used in international speech recognition research related to spoken document retrieval in the broadcast news domain. It must be noted however that the collection is limited in two ways. Firstly, it provides training data for the news domain in particular and secondly, the collection largely contains *written* text data. It has to be acknowledged that as a consequence the data collection is not optimal for modelling the spoken language in the broadcast news domain.

In the next chapter, the normalisation of the raw text data to a format suitable for language model training will be discussed.

Chapter 8

Text Normalisation

Language model training corpora are usually collected in the original format and may therefore contain different character encodings, punctuation marks and a variety of special characters. Moreover, words appear in a many orthographic forms and are sometimes spelled incorrectly. Therefore, these corpora need to be normalised first in order be useful for language model training. This chapter describes the normalisation steps that were undertaken to make the collected Dutch corpora suitable for training.

8.1 Introduction

In the previous chapter the Dutch text corpus for the training of Dutch language models was described. This corpus cannot directly be used for language model training. Due to specific operating systems and/or encoding formats used by the different content providers, the data contain various character representations originating from different encoding schemes, such as Latin1 (ISO-8859-1), HTML and Unicode. The texts are packed with all kinds of punctuation marks, numbers, abbreviations and special characters (such as the %-sign). Furthermore, words appear in a variety of orthographic forms: with or without initial capitals, completely in capitals, with capitals in the middle, with incorrect or missing diacritics and with a wide range of other types of spelling errors. These kinds of text data, referred to as raw text data, first have to be “cleaned” or *normalised* in order to be useful for language model training.

Normalisation is usually not regarded as a very challenging research topic but merely a job that has to be done to enable further research. With the huge amounts of text data used in language modelling, it is even doubtful if an accurate normalisation procedure would significantly improve language model or speech recognition performance compared to more drastic approaches, as the one that just removes all punctuation marks and special characters regardless of the context, ignores orthographic variation and converts all words to lowercase words without diacritics. In spite of this, it seemed worthwhile to discuss normalisation for Dutch language modelling in detail in this thesis, firstly, to give a global impression of the normalisation difficulties encountered in Dutch text data. A second motive stems from the lack of detailed descriptions of normalisation procedures in the literature and the absence[1] of (Dutch) software that can readily be used for the normalisation task, which is surprising given its value for a variety of applications in language and speech technology. By providing and documenting a possible approach to the normalisation problem for Dutch and by implementing the normalisation procedures described below in a publicly available normalisation module, this gap could be bridged. In the next section, the goals of the normalisation procedure in the context of language model training are defined, followed by a description of the subsequent components. Finally the effect of the procedure on the data characteristics is described.

8.2 Normalisation for LM training

Normalisation in the context of language model training is concerned with the reduction of *lexical variability*, or the number of word forms in the word list derived from the corpus, further referred to as lexical items or *lexemes*. Lexical variability is firstly determined by language. For some languages, the *intrinsic* lexical variability is larger than for others. Finnish for example, has a large variety of inflected forms for most of the words (e.g., Siivola et al., 2001) resulting in a huge morphological variability. Word compounding is another language feature that influences lexical variability for languages such as German and Dutch (see also Chapter 9). This type of lexical variability can be addressed by means of stemming or by splitting compound words. However, lexical variability in the context of text normalisation does not refer to this intrinsic, language specific, lexical variability. Instead, variability is addressed that is caused by orthographic variation due to the original function of the text or caused by the actual creation of it, such as the occurrence of punctuation marks, lay-out and formatting markers, and human flaws in a broad sense.

In general, the choice for specific normalisation steps depends on the intended use of the corpus. If it is to be used for part-of-speech tagging, it can be useful to keep the punctuation marks in the data to facilitate the tagging process. Statistical language modelling however is purely based on counts of words (or n-tuples of words) in the training data. Therefore punctuation marks are usually removed as they severely contaminate the counts

[1] Absent in this context means untraceable. From an information retrieval point of view, something that is untraceable, is in fact non-existent provided that the (un)trace-ability is determined on the basis of thorough search efforts. As thoroughness is highly subjective, it must be noted that the literature available to the author and both Internet and human sources were consulted to provide information about existing documentation and software.

’Bovendien:	10	’Bovendien,	27
”Bovendien,”	11	bovendien.	286
’Bovendien’,	11	Bovendien ←	29172
,,Bovendien:	14	’Bovendien	335
Bovendien, ←	1433	,,Bovendien,	34
”Bovendien	181	bovendien,	383
“Bovendien	19	,,Bovendien”,	39
bovendien ←	20089	,,Bovendien ←	719
bovendien:	252	Bovendien: ←	731

Table 8.1: Example of the different versions of the Dutch word “bovendien” with word frequency information, as they appear in a raw text database of 170M words

of single word forms, as illustrated in Table 8.2. Here, 61 orthographically different versions of the Dutch word “*bovendien* (English: moreover)” occurring in a raw text database of $170M$ with a minimum frequency of 10 are listed. The ultimate goal of normalisation for language modelling is to obtain “clean” text versions, containing a representative set of semantically or morphologically distinct words. With a clean text corpus word counts can be used reliably to estimate which words are the N most important words in the corpus domain. This information is used to construct language model vocabularies that are expected to have an optimal *lexical coverage* in a comparable domain. To illustrate the effect on lexical coverage of using raw data, suppose that a $20K$ vocabulary was constructed using the top $20K$ words of the text data, used to create the list in Table 8.2. The word with the lowest word frequency included in this vocabulary, would still have a word frequency of roughly 500. As a consequence, one would end up with five versions (denoted with a ← in the Table) of “*bovendien*” in the vocabulary. Evidently, there is no sense in incorporating these variations. The wasted vocabulary space could better be occupied by (four) other words, resulting in an improved lexical coverage. A comparably disturbing effect of raw data is observed in n-gram training. Not only are the unigram counts unreliable using the text data in the above example, but the bigram and trigram counts, that already suffer the most from data sparseness, are scattered across many lexical items.

As normalisation tools for Dutch were not available at the outset of this research, a set of normalisation scripts were created, primarily tailored for the intended use of the normalised texts: language modelling for speech recognition. In one of the few publications in which normalisation is discussed in detail, Adda-Decker and Lamel (2000) distinguish several normalisation steps that may be deployed, depending on the characteristics of the language being studied. Also, the effect on lexical coverage of different normalisation steps given a French text corpus are described. Of the steps mentioned, especially the processing of punctuation marks, the processing of capitalised sentence starts and digit processing appeared to be very effective in terms of lexical coverage improvement. For the Dutch text database, comparable normalisation steps were applied. The normalisation effects were measured in terms of differences in total words and distinct words, and in lexical coverage improvement as in the French study. Apart from providing these measures, it is almost impossible to provide performance evaluation statistics on the normalisation procedure as a whole, for example in terms of precision and recall. Checking parts of the produced text versions manually is not feasible as the number of errors in a text part of manageable size is too marginal to provide useful statistics about error types. The only way to obtain more information about the performance of a normalisation step is to look into word items with very small frequencies in the word frequency file. When word items have a very low frequency, it is often an indication that there is something wrong with the word or a normalisation step. However, this method was extensively used for debugging and optimisation purposes and therefore not useful for the retrieval of performance statistics.

8.3 Normalisation procedures

In the following, punctuation marks (such as *!*), unit markers (such as *%*), numbers and other characters not in the alphabet-range (such as *#*), will be referred to as *special characters*. Processing these special characters is undoubtedly the most effective step in the normalisation procedure but also the most difficult one. Special characters are the main cause of the large lexical variability in raw text data (as illustrated by the example of “*bovendien*” in Table 8.2), but their function is not seldom ambiguous. Take for example the words “*-’s morgens-* (English: in the morning)” and “*-bovendien’-*” where the quotation mark character has a different function for each word. This ambiguity necessitates a close look at the context of the characters.

Before the normalisation procedure is described in detail, some aspects of normalisation in the context of language modelling for speech recognition that were hitherto passed over without comment, need to be addressed. Firstly, providing for a consistent speech recognition output, for instance by keeping to Dutch spelling rules, runs parallel with the normalisation process. This sometimes conflicts with the lexical variability reduction goal. For example, variability can be reduced by converting all words to lowercase (or uppercase as in some North-American/English systems) or by even removing diacritics as well[2]. But when the output of the speech recognition system is used within an information retrieval framework, as

[2] Note that in Dutch case distinction and, with only a few exceptions, diacritics are not significant for the meaning of words.

is the case in this research, keeping the original case for named entities (such as cities, companies, persons) can be helpful in later (information retrieval related) processing steps (e.g., Kubala et al., 1998). Another example is the processing of hyphens. Removing the hyphen in words such as "*Groot-Britannië* (English: Great Britain)" will decrease lexical variability, but produces speech recognition output that departs from Dutch spelling rules.

Furthermore, normalisation in a speech recognition context requires that special characters are converted to "spoken" forms whereas others can be deleted. An unambiguous conversion example is the "*%*"-sign that is converted to "*procent*" in order to obtain a "spoken" variant. For some special characters the decision to convert or to delete is not evident and depends on either the desired speech recognition output or its intended function in the language model training itself. An example of the latter is the conversion of the period as end-of-sentence mark to "⟨s⟩", to act as a context-cue in n-gram estimation (see Chapter 6).

It must also be mentioned that given the large amount of typographic oddities (formatting errors, spelling errors) in newspaper data, it is extremely hard, if not impossible to process *every* single special character satisfactorily. With a consistently growing data collection, every now and then an ambiguous "special-character-case" pops up that was not yet accounted for and that calls for a decision to be made: ignore it, as its frequency of occurrence is small, or normalise it, which means going through the normalisation algorithms again in an attempt to resolve this without disturbing anything else.

A final remark concerns the order in which special characters are processed. This is important as the normalisation steps are implemented in a modular fashion. Take for example the lexical item "*bijv.*". It is evident that one should solve the abbreviation (expand "*bijv.*" to "*bijvoorbeeld* (English: for example)") before the period is processed in a separate normalisation step.

Recapitulating, the processing of special characters is difficult and solving every single case satisfactorily is almost impossible. In general, for every special character one must determine:

1. what is the intended goal of processing them:
   - decreasing lexical variability
   - avoiding violation of Dutch spelling rules (speech recognition output)
   - create "spoken" variants (language modelling, speech recognition output)
   - creating context-cues for language modelling
2. whether the character is ambiguous or not

Below, all normalisation steps are described in the order in which they were applied. Two normalisation levels are distinguished: a standard level and a “variant reduction” level. The standard level (denoted below as “N” with a number for every subsequent normalisation step) refers to normalisation procedures that have an effect on the total amount of words. The procedures in the variant reduction level (denoted below as “VR” with a number) do not change the total amount of words but try instead to reduce the number of distinct words by mapping different word variants to one single form.

Encoding issues (N0)

As the text data came from different sources (*PCM Publishers* , teletext capturing card, broadcast company, Internet), there was no uniform encoding scheme. The collection contained HTML and Unicode entities and also different character sets were detected. Therefore, the first normalisation step that was taken, was coding the data to an encoding scheme that is adequate for the Dutch language: ISO-8859-1 (Latin1). The resulting decoded text served as starting point for the next normalisation steps.

Brackets and very short sentences (N1)

Single brackets are non-lexical items that should be removed from text data used for language model training. However, as there is usually text in between the brackets, they cannot simply be deleted as by doing so, ungrammatical sentences are created. Therefore, *bracket-pairs* were searched for, in order to remove both the brackets and their content. When, after encountering an opening-bracket, a closing-bracket could not be found within the scope of a paragraph (paragraphs end in a newline character) or before encountering another opening-bracket, only the first encountered opening-bracket was removed.

In newspaper data, very short quasi-sentences are encountered: a date and a place at the start of articles, lines with the name of the author of the article, the photographer of a photo and short titles. As these sentences are not very useful for language modelling and frequently ungrammatical (especially titles), sentences containing four or less words were filtered out.

Abbreviations (N2)

Abbreviations are not encountered in spoken language. Instead, the abbreviated words are (normally) fully pronounced. Abbreviations should therefore be translated to their spoken counterparts in order to be able to model their occurrence in speech. For this purpose, a translation table of 750 Dutch abbreviations was created using a Dutch dictionary so that an abbreviation such as “*z.g.a.n*” could be expanded to “*zo goed als nieuw* (English: as good as new)”.

Quotes (N3)

Quotation marks appear in many forms in the text data and are an important cause of lexical variability. Double quotes are relatively easy to process but the single quote is an example of an ambiguous special character. It should be removed when used as a quotation, but in order to provide for a consistent speech recognition output, should be preserved in the following examples:

(1) in plurals when the singular form ends with a vowel, as in “*foto’s* (English: photos)”

(2) in prefixes, as in “*’s morgens* (English: In the morning)” or “*’45*”

(3) in foreign words, as in “*l’ancien*”

(4) in personifications of acronyms, as in “*NS’er(s)* (English: person who works for the Dutch railway company)”

(5) in names, as in “*O’Connely*”

(6) in possessive forms, as in “*Anne’s boek* (English: Ann’s book)”

(7) in abbreviations, as in “*m’n* (English: my)” or “*A’dam*[3]

(8) in diminutives, as in “*ABC’tje*” (English: small ABC)

Note that in example (4), the single quote was indirectly preserved. Given the relatively high frequency of these cases, the suffix “*’er(s)*” was converted for optimal variability reduction, depending on the character that preceded the single quote:

(A) to “ *_er(s)*” after “*E,A,N,M,F,S,Z,K,R,Y,H,J,L,X*”[4] (*NS’er* → *NS _er*),

(B) to “ *_wer(s)*” after or “*O,U,Q*” (*TNO’ers* → *TNO _wers*),

(C) to “ *_jer(s)*” in all other cases (*DTP’er* → *DTP _jer*).

Three conversions were needed to provide for correct pronunciations of the suffix being “promoted” to a stand-alone word. After a speech recognition run, the original suffixes can be regained in a post-processing step.

[3] *Amsterdam* is sometimes abbreviated as *A’dam*

[4] characters may have diacritics

Commas (N4)

The comma appears between words or between numbers and sometimes a pair of these is used as a substitute for a double quote. In the latter case the commas were deleted. When appearing between numbers they were left alone to be processed later during the processing of numbers (see below). In all other cases the commas were converted to "⟨comma⟩" so that they could serve as context-cues for language modelling. As commas usually indicate a possible phrase-boundary, the position of the commas can possibly be used to add filler words or breath-noises at (some of) these positions to make the text data resemble spontaneous speech (Gauvain et al., 1997).

Various conversions and deletions (N5)

The normalisation carried out in this step all focussed on unambiguous special characters. It concerns question marks, exclamation marks and non-word characters such as "*#*" that have no special meaning for language modeling and could all be deleted, except a few special characters that were converted to "spoken" variants such as "*%*" to "*procent* (English: percent)" and "°" to "*graden* (English: degrees)". For the same reasons mentioned in step N4, the colon and semi-colons were converted to the context-cues "⟨colon⟩" and "⟨semi-col⟩".

Hyphens (N6)

According to Dutch spelling rules, hyphens are required in a number of cases, for example in geographic names ("*Groot-Britannië*"), at word boundaries when vowels ambiguously clash as in "*zee-egel* (English: sea urchin)" and in various other compounds (page 45–47 Woordenlijst Nederlandse Taal, 1995). In Dutch the hyphen can be used to replace parts of compounds when this compound part reappears later in the sentence in another compound. For example, the Dutch phrase "*in voor**spoed** en tegen**spoed***" can be written as "*in voor- en tegen**spoed*** (English: in good and bad times)". As hyphens in words increase lexical variability, all hyphens were removed and compound words were split by converting all hyphens to a space, except when the first constituent was a single character, as in "*e-commerce*". The latter was done because mono-phone words are hard to recognise correctly. By normalising hyphens this way, Dutch spelling rules were regarded as subordinate to the expected substantial decrease in lexical variability.

Periods (N7)

The most frequently appearing ambiguous special character is the period. It can mark the end of a sentence, appears in abbreviations and initials, in WWW related words, in numbers and in constructions such as "*en toen*

. . . niets meer (English: and then . . . nothing)”. In most cases, the period is an important cause of lexical variability. However, as its function cannot always clearly determined by a shallow look at the word and its surroundings, the normalisation of words with periods is difficult. The period that serves as end-of-sentence was detected by looking at the following word that should be more then one character long (otherwise the period could belong to an initial, if the word containing the period has only one uppercase character, see [2] in Table 8.3) and have at least an uppercase first character (see [1] in the Table). In most cases, this algorithm is sufficient for detecting sentence boundaries, although some exceptions had to be implemented, for example when a space was missing after the period (the period appears in the middle of words). In such a case, it was necessary to verify whether the word could be a URL or email address (see [5]). Multiple periods also necessitate special treatment (see [6] and [7]). Periods in abbreviations were processed first, those in numbers were processed in later processing steps.

function	before normalisation	after normalisation
end-of-sentence	*. . . einde. Begin*	*. . . einde* ⟨*s*⟩ *Begin*
abbreviation	*bijv.*	*bijvoorbeeld*
initials	*J. F. Kennedy*	*J*⟨init⟩ *F*⟨init⟩ *Kennedy*
numbers	1.000.000	1.000.000
www	*www.bla.com*	*www punt bla punt com*
not eos	*en . . . toen ging hij weg*	*en toen ging hij weg*
eos	*weg . . . Toen ging*	*weg* ⟨*s*⟩ *Toen ging*

Table 8.2: Normalisation applied to the period

Numbers (N8)

Numbers were expanded to their “spoken” counterparts in order to reduce lexical variability. For example, “1,000” was converted to “*duizend*” and “100,000,000” to “*honderd miljoen*”. Two things need to be noted. Firstly, numbers were written out in a “split” form for optimal variability reduction. So “*21*” was converted to “*één en_ twintig*”. To be able to glue the numbers together again in a post-processing step the underscore was introduced in “*en_*[5]”. Secondly, the dots and commas in numbers were also converted to their counterparts in speech: “12,50” became “*12 komma 50*”. If a number with a comma or dot was preceded by a currency symbol, the dots and commas were replaced by written variants of currency units, so “*fl.12,50*” became “*12 gulden 50*”.

[5] The word “*en_*” is pronounced differently as the regular word “*en*”: /@ n/ instead of /E n/. Note that “1” is converted to “*één*”.

Case normalisation (VR9)

To reduce lexical variability caused by the large amount of case-variants in the text data, all case variants of words were converted to a standard case-variant using word frequency information. It was assumed that the most frequent variant in the text data was the standard spelling form. Although this assumption may not be valid for all cases, it provides a valuable criterion to reduce the number of word variants in the collection. The procedure was as follows:

1. From all available newspaper text data normalised up to this point, a word frequency file was generated. To avoid words at start of sentences (that are always written with a capital) contaminating the frequencies, all words at the start of sentences were discarded.

2. Frequencies of all variants of a word in the word frequency file were compared. The most frequent variant was stored in a conversion table with a "key" form in lowercase as search key and the most frequent variant as standard form. For example, of the word "*groenlinks*[6]" the following variants were found in the data:

   - a version completely in lowercase: "*groenlinks*"
   - a version completely in uppercase: "*GROENLINKS*"
   - a capitalised version: "*Groenlinks*"
   - a version with special capitalisation: "*GroenLinks*"

   As the "*GroenLinks*" variant had the highest frequency, the key "*groenlinks*" with the most probable correct form "*GroenLinks*" was added to the conversion table (see Table 8.3).

3. When variants had an equal frequency of occurrence, no decision criterion actually existed for selection. In such cases one variant was (randomly) chosen in order to achieve an optimal lexical variability reduction.

4. During the normalisation process, words were first converted to the key form and looked up in the conversion table. If a conversion entry existed it was rewritten, otherwise the lowercase base-form was regarded as the standard form.

To ensure that the collected word frequencies are reliable indicators for the preferred spelling of words, it is crucial to use training text data that is expected to have the least amount of inaccuracies. As auto-cue transcripts, text data from the Internet and especially teletext subtitling were expected to suffer the most from spelling inaccuracies, only newspaper text data were selected for this procedure. The case-conversion table that was created eventually using $370M$ words of newspaper text data contained $280K$ entries.

[6] *GroenLinks* is a Dutch political party

key	correct	key	correct
cao	CAO	zeeland	Zeeland
theo	Theo	poolse	Poolse
chirac	Chirac	dow	Dow
n	N	upc	UPC
marokko	Marokko	brabant	Brabant
mun̈chen	München	jorritsma	Jorritsma
ruud	Ruud	poetin	Poetin
denemarken	Denemarken	imf	IMF
canadese	Canadese	arena	Arena
albanese	Albanese	delft	Delft

Table 8.3: Excerpt of case-correction table

Diacritics related spelling correction (VR10)

Diacritics related errors were frequently found in the training data. The word "*café*" for example was found (in a subset of 40M words of the complete training material, see section) 2051 times written correctly, but also 45 times written incorrectly without the acute accent. To reduce the variability for words with accents, again word frequency information and a conversion table was used to solve erroneous spelling variants. The procedure resembles the procedure for solving case variants:

1. From all available newspaper text data, normalised up to this point, a word frequency file was generated.

2. A key form of a word was created by converting characters with diacritics to their ASCII counterparts in lowercase. The variants "*café*", "*cafë*" and "*cäfé*" were converted to "*cafe*".

3. Frequencies of all variants of a word in the word frequency file were compared. The most frequent variant was stored in a conversion table with a key form as search key and the most frequent variant as standard form.

4. When variants had an equal frequency of occurrence, one variant was (randomly) chosen in order to achieve an optimal lexical variability reduction.

5. During the normalisation process, words were first converted to the key form and looked up in the conversion table. If a conversion entry existed it was rewritten, otherwise the key form in ASCII was regarded as the standard form.

The diacritics-conversion table that was created using all newspaper data contained 18K entries. An excerpt is shown in Table 8.4.

key	correct
geneve	Genève
geinformeerd	geïnformeerd
beeindigen	beëindigen
ethiopie	Ethiopië
enquetecommissie	enquêtecommissie
beinvloed	beïnvloed
jordanie	Jordanië
israeliers	Israëliërs
fpo	FPÖ
coordinator	coördinator
cafes	cafés
venetie	Venetië
australier	Australiër
maleisie	Maleisië

Table 8.4: Excerpt of accent-correction table

Various spelling errors (VR11)

Evidently, a lot of spelling errors that are not related to diacritics or case are encountered in the text data as well. Most of these errors cannot easily be corrected as they are often ambiguous. Solving them would require a refined morphological and/or syntactical analysis. A first attempt to correct these spelling errors was done using a spelling correction table provided by the Dutch dictionary publisher Van Dale[7]. The correction table was created on the basis of a list of over a million distinct words encountered in the newspaper texts. The list was processed by the spelling corrections procedures at Van Dale, resulting in a table with the original words in the first column and a varying number of possible correction alternatives in the next column(s). The alternatives were ranked by the amount of effort needed (insertions, deletions, etc) to get to the correct form, the alternatives with the least amount of effort coming first. This correction table could however not be successfully deployed for automatic spelling checking. Correction alternatives are applicable in human controlled applications, as is usually the case with word processors. However, human interference is evidently not feasible for the correction of large text databases. Attempts to make

[7] see also Appendix C

choices out of the word candidates automatically by selecting the "least effort" candidates or by using frequency information, failed. Although many errors could be corrected successfully, it could not be avoided that a relatively large number of errors were incorrectly altered as well. Therefore, this method had to be abandoned.

Some types of spelling errors can be corrected with relatively straightforward procedures. Such errors include characters appearing more than twice in a row (such as in "*onmiddelllijk* English: immediately"). When such character sequences were found, the sequence was truncated so that a sequence of two characters remained. The resulting word candidate ("*onmiddellijk*") was looked up in a word frequency table. If the candidate existed, the word was replaced. Another type of error was seen in compounds of which the final "n" in the first constituent was forgotten or incorrectly included, such as in "*panne(n)koek* (English: pancake)" and "*spinne(n)wiel* (English: spinning wheel)". These errors are relatively frequent due to the combination of relatively complicated spelling rules and a recent change of these rules. To solve most of these errors, word frequency information was deployed. When one variant was encountered, its counterpart was created automatically. Both variants were looked up in the word frequency table. The variant with the lowest word frequency was substituted.

A third type of error is the incorrect use of quotes in plural or possessive word forms. In Dutch a quote is in certain cases inserted before the "s" when just appending the "s" could result in a confusion about the word's pronunciation. This is the case with words ending with a vowel, such as in "*Anne's fiets* (English: Ann's bicycle)" or "*tien foto's* (English: ten photographs)". However, when the word ends with a vowel but the pronunciation does not become ambiguous, the quote is not included, as in "*Gores campagne* (English: Gore's campaign)" or "*tien politiebureaus* (English: ten police stations)". Words with an incorrectly inserted quote could easily be corrected when the character before the quote was a consonant: in that case the quote was removed. When the preceding character was a vowel however, word frequency information was again used. Assuming that in newspaper data the correct spelling form is most frequent, the variant with the lowest frequency in the newspaper text collection was replaced by the one with the highest word frequency.

Finally, a list of words that were undoubtedly correctly spelled (as these came from the Van Dale dictionary) was used to solve errors. The interchange of "c" for "k" or vice versa is a frequently encountered error in Dutch text data. The word "*rekruteren* (English: recruit)" for example is often written as "*recruteren*". Given both variants, the one that was not found in the correct word list was replaced with the one found.

Acronym processing (N12)

In 360*M* words of newspaper data more then 8500 acronyms were observed. About 1000 of them appear in the top 65*K* most frequent words. In Adda-Decker and Lamel (2000) acronyms are split up into separate characters ("*ABCD*" becomes "*A. B. C. D.*") to reduce lexical variability. A disadvantage of this acronym splitting procedure is that all separate characters must be recognised correctly in order to retain the acronym in a speech recognition task. Given the small amount of acoustic information in single characters, recognising these correctly is very difficult. On the other hand, although single-character recognition errors are circumvented when acronyms are not split up, recognition errors will be introduced for acronyms that are not included in the recognition vocabulary. A compromise was found in leaving the 1000 most frequent acronyms untouched and splitting up all other acronyms that were encountered. This way, lexical variability can be reduced whilst the most frequent acronyms will not suffer from incorrect character recognition, and less frequent acronyms still have a chance of being recognised as a sequence of single characters.

8.4 Normalisation results

The effect of the normalisation procedures was evaluated by looking at lexical variability reduction. In Table 8.4 the changes in total number of words, number of distinct words, and the self-coverage of 65K lexicons derived from the normalised text versions are shown. Also, the ratio of the total number of words and the number of distinct words are given, as the ratio metric is indicative of how well the *n*-gram probabilities can be estimated. The 'N⋆' procedure shows the results for the alternative VR9 and VR10 normalisation procedure: converting all words to uppercase words without diacritics. The absolute differences in unique words relative to the previous step are given in the "Diff" column. The labels in the left column correspond with the labels on the x-axis in Figure 8.1 on page 121 that plot the statistics of Table 8.4.

8.5 Discussion and conclusion

The normalisation procedures show a 10.5 % increase in the total number of words and more importantly, a 64 % decrease in distinct words. Lexical coverage consistently increases, reaching a 16.33 % relative gain. So, lexical variability has been reduced substantially. Especially steps N3, N4, N6 and N7 have contributed a great deal to this reduction. As the ratio metric also increased, from 29 to 93, the robustness of the *n*-gram models that are trained using the normalised corpus, is expected to improve.

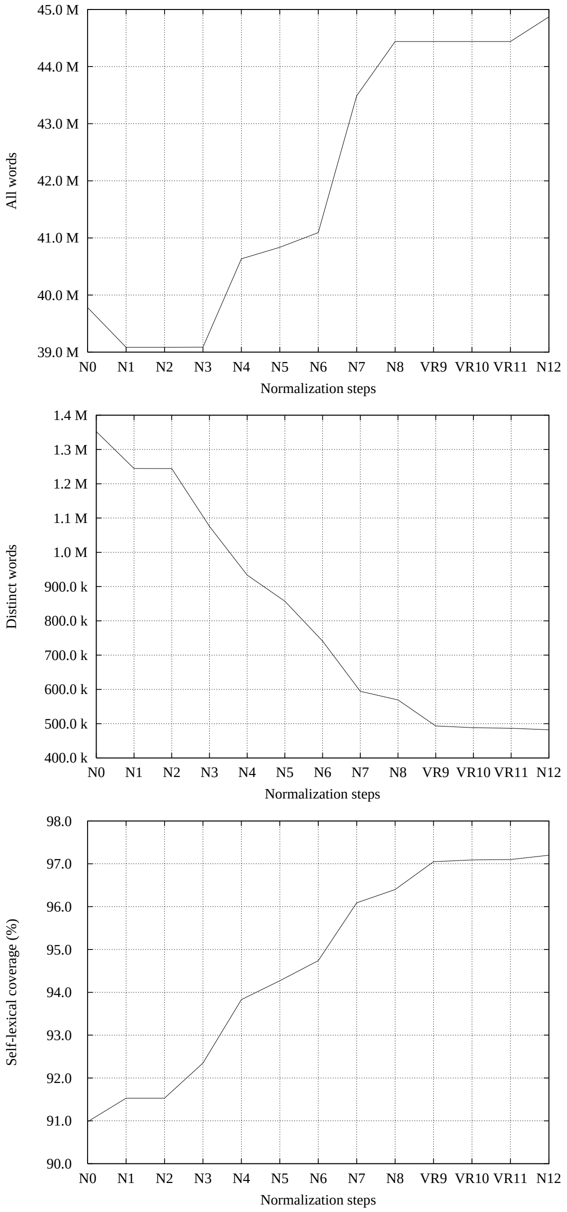

Figure 8.1: Effect of normalisation steps on total number of words (upper-left), number of distinct words (upper-right) and lexical coverage (bottom)

	Total words	Unique words	Diff	Lex cov	Ratio
N0	39,780,228	1,352,167		90.98	29
N1	39,085,707	1,244,624	13,971	91.53	31
N2	39,085,730	1,244,383	-241	91.53	31
N3	39,088,288	1,075,832	-168,551	92.35	36
N4	40,634,465	933,539	-142,293	93.83	43
N5	40,836,087	857,188	-76,351	94.27	47
N6	41,093,448	741,116	-116,072	94.74	55
N7	43,488,975	594,353	-146,763	96.09	73
N8	44,437,768	569,158	-25,195	96.40	78
N⋆	44,437,768	487,906	-81,252	97.25	91
VR9	44,437,768	493,453	-75,705	97.05	90
VR10	44,437,768	488,162	-5,291	97.09	91
VR11	44,437,768	486,634	-1,528	97.10	91
N12	44,869,776	482,197		97.20	93

Table 8.5: Effects of the normalisation procedures on the number of words, distinct words, their ratio and the (self) lexical coverage of a 65K lexicon. The names in the left column correspond with the names on the x-axis in Figures 8.1 on page 121. The N⋆ procedure shows the results for the normalisation procedure that converts all words to uppercase words without diacritics.

A few remarks need to be made. Firstly, the expansion of abbreviations hardly had an effect on lexical variability. It reduced the number of distinct words by only 241 words. Most of the abbreviations were apparently expanded to single words (such as "*bijv.*" to "*bijvoorbeeld*") given the small increase in the total number of words due to the abbreviation expansion step. Secondly, it must be noted that the huge increase in total number of words after the processing of the comma and the dot was caused by the conversion of commas and dots to the context-cues "⟨comma⟩" and "⟨s⟩" respectively. However, as commas and dots were often attached to words, detaching them reduced the number of distinct words dramatically. Thirdly, of the variant reduction (VR) normalisation steps, especially the case normalisation step substantially reduced lexical variability. The effect of the error correction normalisation steps (VR10 and VR11) was evidently much smaller (errors are less frequent than case differences) but as the distinct words were reduced with 7000 words, it means that a satisfying amount of at least 7000 misspelled words could be corrected. Finally, note that when the case normalisation and diacritics normalisation steps are replaced by a single procedure that just removes all case and diacritic related distinctions by converting all words to upper-cases without diacritics, the lexical variability reduction is not exactly the same. This is due to the few words that have a variant with and without diacritics, such as for example "*één*

(English: one)” and “*een* (English: a)”.

In principle, all topics related to language modelling that are addressed in the following chapters, use text versions that are normalised as described above. However, it must again be pointed out that the normalisation process cannot be regarded as a non-recurring procedure. Instead, due to a growing data collection or a shift of focus on the data, new cases pop up that necessitate normalisation. Illustrative is the spelling of the word “Qaida”. In older newspaper material it was usually spelled as “Kaida”. But this became only apparent during the evaluation of speech recognition runs on recent material, showing that “Qaida” in the reference transcript was sometimes substituted by “Kaida”. The speech recognition vocabulary unnecessarily contained both versions due to their frequent occurrence in the training data: “Kaida” in older material and “Qaida” in more recent material. When cases like these showed up, the normalisation procedures were adapted wherever possible. Isolated cases that could not be fit into existing normalisation steps were listed in an exception table and their occurrences in the data were normalised individually.

Chapter 9

Vocabulary optimisation

An appropriate word selection for the language model vocabulary is crucial for an optimal coverage of the words appearing in the task domain. In this chapter, the selection of words for the language model vocabulary is addressed. A novel word selection method is introduced that incorporates long-term temporal information in the word selection procedure. This method and other vocabulary selection methods are compared in terms of out-of-vocabulary rate in a simulated longitudinal broadcast news transcription task.

9.1 Introduction

Constructing a vocabulary is a crucial preparatory step in statistical language modelling (LM) for large vocabulary speech recognition (LVCSR). Its quality contributes a great deal to the quality of the model and ultimately, ASR performance. The better the vocabulary covers the words in a task domain, the less the speech recognition will suffer from *out-of-vocabulary* (OOV) words that are an important source of error in speech recognition systems. In speech recognition the coverage of a vocabulary is usually expressed in terms of *lexical coverage*, the ratio between the number of words in the task domain that are in the vocabulary and the total number of words in the task domain. The OOV rate is its counterpart and is defined as the ratio between the number of words that are *not* covered by the vocabulary and the total number of words in the task domain. As a speech recogniser substitutes OOV words for the most probable alternatives given the acoustic model and language model, the N-gram language model probability of the *next* word is computed on the basis of an incorrect word. In other words, the language model probability of the word following the OOV word is based upon a “corrupted” history. Therefore, an OOV word may not only result in one word not being recognised correctly but instead, could damage the recognition of the next word(s) as well. Several studies (Gauvain et al., 1995, e.g.,) showed that one OOV word can result in between 1.2 and 2.2 word recognition errors.

From an information retrieval (IR) point of view, the vocabulary of the speech recognition system can be viewed at as the set of words that will be used to create *representations* of the documents in the task domain. For a successful retrieval of the individual documents the vocabulary must cover the words in these documents well. OOV words may result in OOV *query* words (QOV): words that appear in a user's query and also occurred in the audio document but, as they were OOV, could not be recognised correctly. OOV's damage retrieval performance in two ways: firstly, given a query with a QOV word, the QOV word leads to a word *miss* in searching. Secondly, its replacement potentially induces a *false alarm* for other queries. Document expansion and query expansion techniques (see Chapter 2.2 on page 32) are often deployed to compensate for QOV's in information retrieval (Woodland et al., 2000, e.g., see). Nonetheless, reducing OOV words in a speech recognition task is worthwhile. All the more as the mentioned expansion techniques keep their added value with respect to retrieval performance, even when a speech recognition system produces only a small number of errors.

In the next sections, a number of vocabulary creation strategies are discussed. These strategies are particularly evaluated on their ability to select an appropriate vocabulary for the broadcast news domain. First, a standard vocabulary selection procedures is described. Next, vocabulary selection methods that use temporal information to reduce OOV words in the broadcast news domain are discussed (Section 9.3), followed by the description of a novel approach to incorporate temporal information in the vocabulary selection procedure (Section 9.4). Finally, in Section 9.5, all described vocabulary selection methods that focus on temporal information are compared in terms of OOV rates in a simulated longitudinal broadcast news speech recognition task.

9.2 Standard vocabulary selection

The difficulty with vocabularies in speech recognition is that their size is usually limited. Evidently, the number (N) of selected vocabulary words influences the representation quality of a vocabulary. The larger N is chosen, the better the representation will be. This is illustrated in Figure 9.1 that plots the coverage of vocabularies as a function of vocabulary size given a newspaper corpus of 300 M words. This corpus was both used for generating the vocabularies by selecting the top N most frequent words, and for measuring lexical coverage. In the context of broadcast news transcription however, one must recognise that as new words (especially names) are introduced frequently, OOV words cannot entirely be banned, regardless of the vocabulary size. Moreover, when a vocabulary of a speech recognition system grows, acoustic confusability becomes more probable as the number of words that differ only in a few phones grows with it. Because of this acoustic confusability the optimal vocabulary size for large vocabulary tasks is estimated to be roughly between $55K$ and $110K$ words Rosenfeld (1995). Another reason for restricting vocabulary size is that speech recognition systems are often bound to a vocabulary size limit of 65536[1] words due to implementation decisions, as is the ABBOT speech recognition system that was used for this research. As most of the research effort in speech recognition used to be concentrated on English, this size limit was not much of a problem: a $65K$ vocabulary often provided near full coverage in English (see Table 9.2 below).

	EN	IT	FR	**NL**	GE
#words	37,2M	25,7M	37,7M	**37M**	36M
#distinct	165K	200K	280K	**462K**	650K
ratio	225	128	135	**80**	55
5K coverage	90.6%	88.3%	85.2%	**84.02%**	82.9%
20K coverage	97.5%	96.3%	94.7%	**92.64%**	90.0%
65K coverage	99.6%	99.0%	98.3%	**97.15%**	95.1%

Table 9.1: Comparison of languages in terms of number of distinct words, ratio and lexical coverage for different vocabulary sizes. The data was borrowed from Adda-Decker and Lamel (2000), except for the Dutch data.

Seymore et al. (1997) estimated a vocabulary in the range of $40K$ - $60K$ to be appropriate for the English Hub4 broadcast news task. That the optimal vocabulary size in this domain may be higher for other languages was illustrated by the study of Adda-Decker and Lamel (2000). In this study languages are compared in terms of lexical variety, lexical coverage and the ratio:

$$\frac{\#\text{words in the data}}{\#\text{distinct words in the data}}$$

that provides an indication of how well statistical language models can be estimated given a certain language. In Table 9.2 the statistics found in Adda-Decker and Lamel are given and those for Dutch (NL) are added. It shows that Dutch is comparable with German (GE): both languages have a low ratio statistic compared to the other languages which means that the training of robust LM parameters is relatively difficult. Lexical coverages are significantly lower: those of German lexicons being even lower than those of Dutch. The reason for the latter is case declension for articles, adjectives and nouns, which dramatically increases the number of distinct words in German. The major reason for the poor ratio and lexical coverage

[1] Using 16-bits integers allows for the representation of $2^{16} = 65536$ words

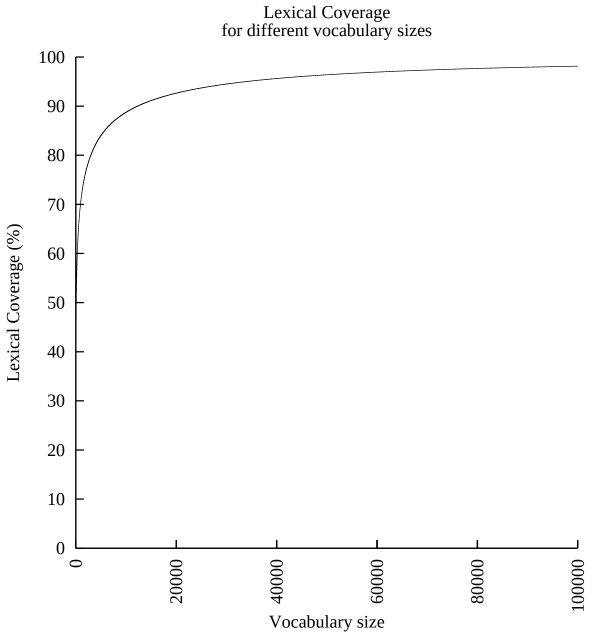

Figure 9.1: Lexical coverage as a function of vocabulary size. Vocabularies were obtained using the top N words in a $300\,M$ newspaper corpus. Lexical coverages were measured using the same corpus.

of German and Dutch compared to the other languages is *word compounding*: words can (almost) freely be joined together to form new words (for example, the valve cap of a bicycle tire is translated as "*fietsventieldopje*" in Dutch). Because of compounding in German and in Dutch, a larger lexicon is needed to achieve the same lexical coverage as for Italian, English or French. As vocabulary size in large vocabulary speech recognition is usually limited to 65K words–so practically invariable–vocabulary space for these languages may be regarded as particularly sparse. For languages such as Dutch or German it is even more important to select only those words that are expected to appear in the task domain in order to use the sparse vocabulary space economically. Another, or complementary option to reduce the number of distinct words in Dutch is to split compounds into their parts. This option will be addressed in Chapter 10.

For creating an optimal vocabulary that is within the limits set by the large vocabulary speech recognition system, development corpora that are similar to the task domain are typically deployed to acquire word frequency statistics that can be interpreted as word unigram probability estimates. If this data is chosen well, normalised appropriately and large enough to obtain reliable frequency statistics, ordering the estimates and selecting the top N words provides a useful word usage representation of the task domain as the most frequent words are also expected to be the most important words in the domain. Note however that having selected the most important words of the domain in the LMs vocabulary does not guarantee an optimal LM performance. As discussed earlier, the eventual LM performance highly depends on the amount of available data to train the LM parameters for the selected vocabulary. N-gram statistical language modelling has proven to work very well in speech recognition, provided that there are sufficient amounts of data to train the word models. Consequently, a very well-fitting domain representation may still produce a low quality LM when representative data for robust word model training is only marginally available. This may occur when for instance a relatively small number of manual transcripts that have a high resemblance with the task domain are used for word selection. The quantity of this type of data may be too small for robust n-gram estimation. Training the LM parameters with large amounts of available off-domain (e.g., newspaper) data instead, may result in unreliable estimates of the n-grams that are not very well represented. This *data sparsity* in language modelling can be avoided by using mixtures of language models, for example one that closely matches the target domain and one trained on the full set of development data (Clarkson and Robinson, 1997). In Chapter 11, the application of mixture LMs will be addressed in more detail.

For certain task domains it may be particularly difficult to obtain an appropriate amount of example data to obtain reliable word frequency statistics for word selection. An illustrative example of such a domain is the "historical archives" domain in the ECHO project (see Appendix A). To be able to include historical names and events, or to retrieve (ancient) word usage statistics, historical text corpora are needed. Such corpora are however not available digitally[2]. In this thesis however, the main focus is on the broadcast news (BN) domain. Although the amounts of Dutch training data for this domain are not as large and diverse as for the English language, the available Dutch newspaper collections provide word frequency statistics that are reasonably close to the BN domain. A number of data selection techniques have been proposed to make the best use of the newspaper data in language model training in the BN domain. In the next sections, these techniques are discussed.

9.3 Vocabulary selection using temporal information

As large amounts of text data that perfectly match the target domain are usually not readily available, data selection often consists of partitioning the available training data, which is usually newspaper data, a source that can relatively easily be obtained. Parts of the data that match a task domain more closely then others are taken together to serve as input for the vocabulary selection or LM training routines. The degree of matching the task domain may for instance be based upon the content of documents in the collection. When documents in the newspaper collections are labeled with topic information (e.g. with human annotations or section information such as “sports” or “business news”, see also section 7), these labels could serve as selection criteria, or reversely, as criteria to *exclude* certain content, such as for example “television broadcast information” that will generally not contain relevant words for the BN domain. When such labels are not available and manually partitioning the collection is not an option, the collection could be partitioned automatically by means of content-based document clustering and classification, for example by using information retrieval techniques (see Section 2.2). The use of information retrieval techniques for the creation of domain specific language models is discussed in more detail in Chapter 11.

Another important information source in newspaper collections that is often easily available is temporal information. Rosenfeld (1995) showed that using temporal information as data selection criterion decreases OOV rates. Lowest OOV rates were obtained using only a relatively small but *recent* portion of the available development data. The quality of the data, expressed here in terms of “recency” appeared to be more important than the quantity of the data. In real time-longitudinal TDT (Topic Detection and Tracking, see also Section 2.4) types of tasks, for example the daily recognition of broadcast news shows, applying recency as a selection cri-

[2] For the ECHO project attempts were made to scan historical documents for vocabulary selection and language modelling purposes. These were however not successful due to the low quality (carbon copies) of the paper prints.

terion is crucial. The BN domain is subject to a constantly changing focus. Words that are frequently used (so apparently important) this week, may not occur at all next week. An illustrative example is the word "*poederbrief*" (English: a letter containing, possibly poisonous, powder). This word was used frequently for a few weeks after 11 September 2001 but after a relatively short while it was hardly mentioned again. "Recency-sensitivity" especially applies for named entities: they suddenly appear and after a longer or shorter while, disappear again. Only a few (presidents, important cities or companies) are more recency-robust and can be spotted over a longer period of time. Therefore, in this type of tasks the vocabulary has to be *updated* regularly to prevent it from being obsolete. As named entities are significant keywords in a (spoken document) retrieval framework and are likely to appear in a query, improving the recognition rate of named entities implicitly means improving retrieval performance. Figure 9.2 illustrates word history information of a few frequent words. The relative frequencies of occurrence every week of these words in the Dutch news database (see Section 7) spanning the period January 1999 until September 2002 are plotted in time. Relative frequencies were taken to normalize for different amounts of newspaper data per day. The plot in the upper-left corner shows the changing of importance of Bush and Clinton after the presidential election. The upper-right one gives an example of words with a strong periodicity: the word "*Kerstmis*" (English: Christmas) and "*Sinterklaas*" (English: Santa-Claus). Below, two examples of single words that gain and loose importance due to news events: the word "*Islam*" (left) and the word "*poederbrief*" both suddenly going sky-high after the terrorist attacks on the 11th of September 2001. The use of the word "*Islam*" however stays at a consistent level whereas "*poederbrief*" sinks into oblivion.

In line with the findings of Rosenfeld (1995), Auzanne et al. (2000) proposed the regular adaptation of the language model vocabulary (and word models) to minimize OOV words in longitudinal speech recognition tasks. These so called "*rolling language models*" (RLM) are created by adding new words to the vocabulary that were seen within a specific lookback time window (*e.g.,* one day) in a parallel news-wire text corpus and removing ("forgetting") ones that were not seen there within another time limit (*e.g.,* 28 days). Only words that had a minimum frequency of occurrence (4) per day were added. Applying a rolling language model on the entire TDT-2 corpus (Cieri et al. (1999)) provided a 22.44% relative reduction in OOV words. Although this procedure seems to work fine given the substantial reduction in OOV's, the procedure is limited to the extent that it adds only those types of words it can "recognize" as new, namely those words that have a minimum count in the chosen time window. In a first version of the procedure, Auzanne et al. (2000) chose a time window of one day. Words that appeared only a limited number of times per day, but appeared with a consistent frequency, could not be picked up by the algorithm. Therefore an alternative frequency-based method was proposed: words were added also when they appeared a minimum count of days within a frequency window.

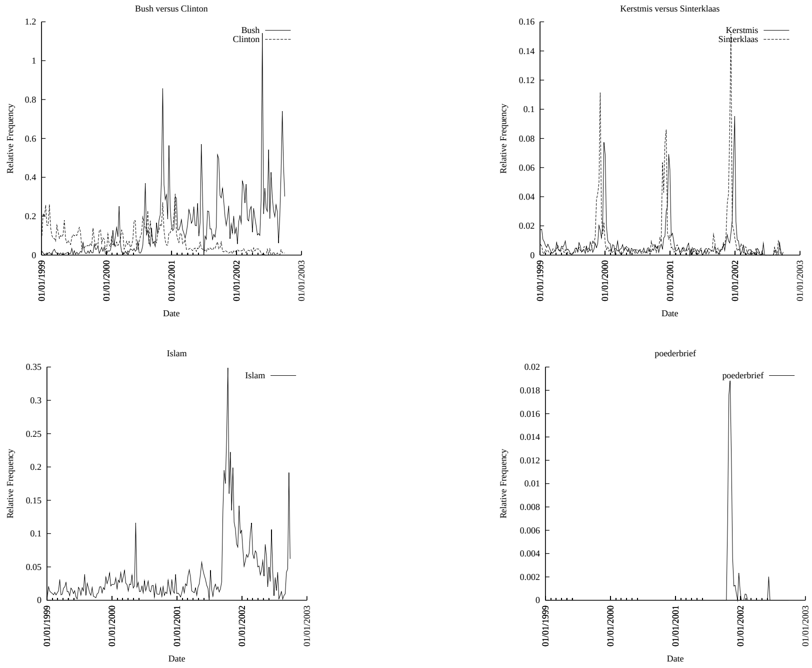

Figure 9.2: Relative frequency statistics plotted in time of the words "Clinton" and "Bush" (upper-left), "Christmas" and "Santa-Claus" (upper-right), "Islam" (lower-left) and "poederbrief" (lower-right). Note that the y-axises are scaled differently.

With a frequency window of 28 days and a count of 4 the lowest OOV rates were obtained.

9.4 Vocabulary selection using binary prediction

The results of the studies mentioned above show that by using temporal information, better language model vocabularies can be constructed for recency-sensitive task domains. By selecting recent training data, some of the word history information as illustrated in Figure 9.2 can be captured so that word occurrences in the task domain can be predicted more accurately. These approaches may be regarded as correction mechanisms for a selection method based on global word frequency information, either by narrowing down the time window that is used to select the vocabulary words (Rosenfeld (1995)) or by taking the global and recent word frequency information as separate information sources (Auzanne et al. (2000)). It can be argued, however, that these approaches have a number of disadvantages. A first disadvantage is that a number of parameters need to be set experimentally (time-windows, word frequency thresholds). News can be highly fluctuating and as a consequence, word frequency dynamics observed in a training collection may differ from those in the target data. Furthermore, in the context of broadcast news transcription, it is questionable whether these parameters should be trained on broadcast news example data, or, to obtain a more global impression of word frequency dynamics, on a general news source, such as newspaper data. Another disadvantage is that these approaches do not take *long term temporal information* into account but instead focus on a relatively short recent time period. However, long term information can be very useful for the selection or rejection of possible vocabulary word candidates as will be explained below.

Using short term temporal information as a correction mechanism is useful to enable the selection of words that would be missed when only global word frequency information was taken into account. Global word frequency information can be viewed as a measure of general "importance" of a word in a certain time-window and in a specific domain. Both time window and domain are determined by the training data selection. Having built up a certain degree of general importance (word frequency), words enter the upper regions of a ranked word frequency list and often literally get stuck in there: it takes some time for new words building up enough word frequency and cast such words off their position. Either by preventing words to build up too much word frequency (narrow down the time window), or by leaving some room in the vocabulary for recent words (separate word frequency lists), new or recent words can be given a better chance to be selected. However, the first approach may result in an under-estimation of general word "importance" if a word happens to occur infrequently in the chosen time-window. This may especially occur for words with a strong periodicity, such as "Santa-Claus". The second approach has the disadvantage that the global word frequency list remains “contaminated” with words that were very important in the past but rarely occur anymore. Names especially can be contaminating with this respect. Take for example names of politicians that appear frequently in the news. When politicians resign, only a few keep a certain degree of general “importance”. The names of the less fortunate ones, may contaminate the word frequency list for a long time. The same applies for words that represent certain events that received a burst of attention for a relatively short period of time. An example is the previously mentioned word “poederbrief (English: letter containing possibly poisonous powder)”: it built up a very strong word frequency count during a short period of time but appeared very rarely after this period.

Supposing that long term word history information as illustrated in Figure 9.2 could be captured into a usable metric, instead of applying *ad hoc* correction mechanisms to word frequency information, long term temporal information could really be incorporated in the selection process. Applying such a selection procedure to the examples in the figure, would result in “Clinton” going down in the rank order of candidates to be selected, although it may keep a position with a high selection chance as “Clinton” still regularly occurs in the news. Likewise, “Santa-Claus” and “Christmas” would rise or go down in rank order in parallel with approaching or receding from the time period of maximum frequency. Finally, “Islam” would rise because of its recent increase in word frequency and for the opposite reason, “poederbrief” would go down.

It is however difficult to capture long term word history information into a single metric. But as the newspaper data was stored per day in the text database that is used for this research, word vectors containing daily word frequency information for a certain time-window could easily be generated (see Figure 9.3) and serve as a starting point in the search for a suitable metric. A number of strategies were considered. One of the first approaches that came to mind was using the vectors as word history *type* representations. Using a clustering algorithm, similar vectors can be grouped together to represent a word type. These types could *post hoc* be characterized such as having a strong periodicity, having a certain degree of decaying word frequency (word is rapidly or slowly losing importance) or, the opposite, a rising word frequency (a word’s importance is growing). These type definitions could then be used in the vocabulary word selection procedure. However, such a method would become rather complex given the relatively large word vectors, a clustering or classification procedure that has to be repeated regularly to assign a type to new words and the necessity to relate the word types to a desired behavior in the vocabulary selection procedure.

Computing a *running average* over the vector, seemed a more workable method. The average, either weighted or not, could then be used as a prediction value for words occurring immediately following the chosen time window (typically tomorrow, supposing that the data of today is available). Unweigthed, a running average represents the number of times a word may be expected to occur on the target day. Using a weighted average, recent occurrences of words could be counted more heavily than occurrences in a more distant past. A drawback of this method, however, is that the running average should preferably be computed differently depending on the word history type. For a word that is loosing importance, an unweighted average may be the most appropriate so that old frequencies still add some value to the average. For a word that is gaining importance on the other hand, the opposite produces the intended result: recent word frequencies must add more weight to the average than old weights to enable its selection for the vocabulary.

The running average method tried to incorporate both temporal information and word frequency information directly into a single metric. As an alternative, the running average could be used to obtain temporal information only and to merge this information with general word frequency information in the domain. The (relative) word frequency information could then be interpreted as a probability measure for a word ω occurring in a specific task domain D_ω, denoted as $P(\omega|D_\omega)$. The temporal information must then provide a measure for the probability that a word ω occurs at a specific point in time given its history H_ω, denoted as $P(\omega|H_\omega)$. The probability that a word occurs in a specific domain at a specific point in time $P(\omega|D_\omega, H_\omega)$ can finally be obtained by combining the two probabilities.

To obtain a metric containing temporal information only, the running average procedure was slightly modified. Instead of incorporating word frequencies as values in the word vectors, binary word occurrences served as vector values: either a 0, when a word did not occur at a specific day, or a 1, when a word occurred at least once a day (see Figure 9.3). This procedure resulted in a long binary vector that was summarized in a so-called *binary prediction* metric (B) by adding all binary counts and dividing this count by the total number of days since the first day that a word occurred. For instance, when a word occurred for the first time two days ago and was not mentioned yesterday, it received a B value of:

$$B = \frac{(H_0 - 2) + (H_0 - 1)}{T} = \frac{1 + 0}{1 + 1} = 0.5 \qquad (9.1)$$

This "first-occurrence-count" was applied to remove the effect of time win-

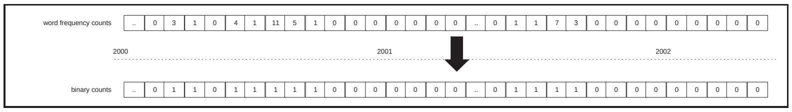

Figure 9.3: An example of a daily word frequency information of a word represented in a vector for a certain time window (top) and the binary version of this vector (bottom).

dow length on the prediction of new words. As soon as new words appear, they will have a high prediction value that quickly decays when their occurrence is only temporary. When the occurrences of new words remain constant, the prediction values will remain constant as well. The binary prediction metric is summarized in equation 9.2. Note that instead of using days as interval for measuring binary values, longer intervals can be chosen. In this way short term periodicity of words (such as sports related words that especially occur on Mondays) can be smoothed out.

$$B(\omega_i) = \frac{\sum_{t=0}^{T} b(w_i)}{H} \tag{9.2}$$

where b is the binary value of word w_i, T the time-window and H the history of the word counted in days from the first non-zero value in the binary vector onward.

Assuming that the binary prediction metric can be used as a representation of the temporal word history information, this metric must be combined with the word frequency information (relative word frequencies) to obtain a measure that can be used for the selection of words for the vocabulary, denoted as $S(\omega_i)$. As both information sources are highly dependent, it is difficult to combine them in a probabilistic framework. Therefore, a relatively simple merging function was chosen that consists of a weighted multiplication of the information values. This choice was based upon the following considerations. Recall that the goal of merging the two information sources was to provide a measure that reflects both a general and temporal "importance" of words in a domain at a specific point in time. Intuitively, when a word has a high general value (high word frequency) and a low temporal value (low binary prediction value) this should be reflected in a decrease of the selection measure $S(\omega_i)$. In the opposite case the selection measure should increase. To model such a behavior, merging the two sources by multiplication was an evident choice. However, as the binary prediction values are proportions between 0 and 1 and the influence of the multiplication on the final result value could be to small to obtain the desired effect, a weight option was added to the binary prediction value in the merging function (equation 9.3):

$$S(\omega_i) = B(\omega_i)^{1/\alpha} \cdot F(\omega_i) \tag{9.3}$$

where $F(\omega_i)$ is the relative frequency of word ω_i, $B(\omega_i)^{1/\alpha}$ the binary prediction of word ω_i and α the weighting factor of the binary prediction value.

In the next section the performance of this method is evaluated by comparing it with a number of vocabulary selection methods. The method described above will further be referred to as the binary prediction method.

9.5 Temporal word selection methods compared

To investigate the effect of different temporal word selection procedures on Dutch data a longitudinal experiment was done that computes OOV words in (simulated) Dutch news broadcasts, given lexicons produced using the selection methods described earlier. The goal was to investigate which selection method is the best method for a broadcast news transcription task for the Dutch language.

9.5.1 Experimental design

Using different vocabulary selection methods, vocabularies containing $65K$ words were created out of newspaper text data. These vocabularies were evaluated by computing out-of-vocabulary rates on broadcast news transcripts of the Dutch "*NOS acht uur journaal* (English: Eight o'clock news)" in the period January 2002 - September 2002. As it was not feasible to generate transcripts of all news shows in this period manually, teletext and autocues were used instead. Together these provide a realistic representation of the word usage in the news shows (see also Section 7.3). All transcripts of one week were taken together (Monday to Sunday) and weekly out-of-vocabulary rates were computed, resulting in 35 OOV rates for each vocabulary selection method. For one week (the week preceding the 5th of May) no OOV rates were computed as teletext data for that week was completely missing from the database[3]. On average every week contained $23K$ words. In Figure 9.4 the total number of words per week are plotted. OOV rates of the vocabularies will more or less follow this curve as the OOV rate will generally be lower when there are less words that need to be covered. Besides the number of words, the actual content of the broadcast news shows in the particular weeks will also influence OOV rates. Note that in this experiment the focus is not on the OOV performance fluctuations in time but on the differences between selection methods.

Vocabularies were created using the following word selection strategies:

1. **Based on static word frequencies**:
   A simple and widely used vocabulary selection strategy entails selecting the most frequent words from a training corpus spanning a fixed time-period and using this vocabulary for speech recognition in a longitudinal task without making any adaptations. Such vocabularies are further referred to as *static vocabularies*. To evaluate the performance of such a selection method, four static word frequency files were created using one year of data in a distant past (1999, referred to as S99), data of the past three years (1999-2001, S9901), the last year (2001, S01) and the last half year (last six months of 2001, S01L). The distant past word frequency file was created to serve as a reference: it repres-

[3] Teletext capturing was off-line that week

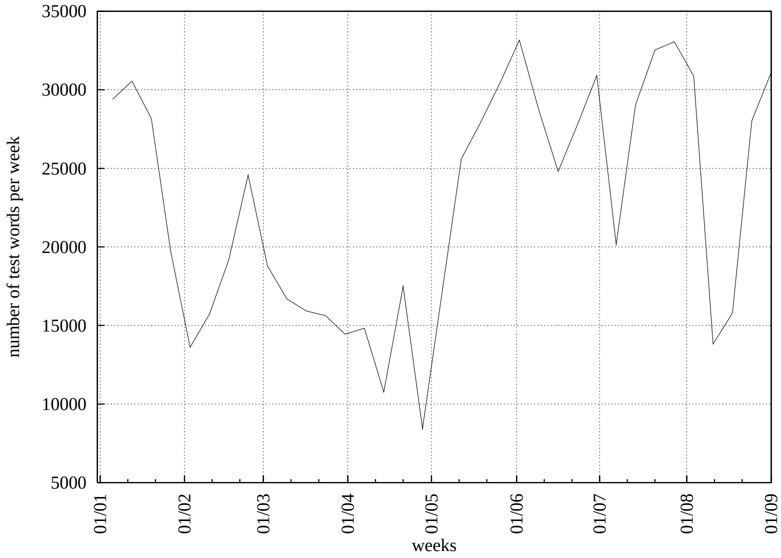

Figure 9.4: Number of test words (y) from teletext and autocues of broadcast news shows (NOS 8uur journaal) for every week (x).

ents a vocabulary that is very much out-of-date. The other word frequencies can be viewed as based on a *look-back time window* of half a year, one year and three years. Vocabularies were created by taking the top $65K$ most frequent words. To give an idea of the number of words that were used to create these vocabularies, the 1999-2001 word frequency file was based upon 255M words of newspaper text data and contained $1,3M$ distinct words. The self-coverage of the top $65K$ words was 97.08%.

2. **Based on open word frequencies**:
Instead of using static word frequencies, word frequency counts can simply be updated with a certain interval (for example every day or week). Suppose that the initial word frequency file was created using one year of training data, after six months of updating, the word frequency file describes one and a half year of data. In this sense, the word frequencies are "open". The vocabularies based on such updated word frequency counts are therefore further referred to as *open vocabularies*. Although this may not have a large effect on the ranking of the words in the word frequency file, this method will slowly force recently and frequently used words to a position in the word frequency file that allows selection for the vocabulary. One open word frequency file was created based on the word frequency file of 1999-

2001 (O9901). The frequency counts were updated every week.

3. **Based on closed word frequencies**:
One can argue that using a long term word frequency file, either static or open, to select words for a vocabulary has disadvantages. Words that have built up a high frequency in the past, but should not be regarded as important words anymore, will stay in the vocabulary for a long time. To avoid that these types of words being included in the vocabulary, the text data's time-window could be narrowed down. However, this time-window should on the one hand be large enough to obtain reliable word frequencies of the domain and on the other hand, short enough to capture recent words successfully. To investigate if such a method could improve OOV rates and what the optimal window size is, three so called *closed vocabularies* were created, one based on a word frequency time-window of approximately half a year of data (168 days, CL168), one based on approximately a year (352 days, CL352) and one of one and a half year (520 days, CL520). The last week that was incorporated in every word frequency file was the one preceding the test week. Every week the time-window shifted one week further, including the most recent week and removing the most outdated week.

4. **Based on rolling language models**:
As an approximation of the RLM procedure described in Section 9.3, for every week a new vocabulary was created using the following procedure. The first 30K words of the vocabularies were taken from the top 30K words from a master word frequency file based on the 1999-2001 newspaper text data. Next, words collected on the past 7 days (the look-back window) with a minimum relative frequency of $R = 0.01, 0.025, 0.005$ (referred to as R01, R005 and R0025 respectively) were added to the vocabulary. Instead of absolute counts, relative frequencies were computed to normalize for different amounts of data in the time-window. Words that were added before but did not appear during the past 28 days (forgetting window) were removed. When there was still vocabulary space left after adding new words, the most frequent words from the master vocabulary that were not already included were added up to a vocabulary size of 65K words.

5. **Based on binary prediction**:
Different versions of the binary prediction method as described above were implemented, varying the length of the time-window and the weight of the binary prediction value. First, vocabularies were created that used the binary prediction values only, so without merging them with word frequency information, to rank the word lists. One was *open* and uses a binary prediction vector based on the period 1999-2001 that was updated every week analogous to the open word frequency method. Another was *closed* and uses a binary prediction

vector based on a closed time-window of 380 days, well over a year to enable the capturing of yearly periodicity. This procedure was repeated, this time however using both word frequency information and binary prediction values, resulting in open binary prediction vocabularies and a number of closed binary prediction vocabularies that differ in the used time-window: 190 days (well over half a year), 380 days (well over a year) and 570 days (well over one and a half year). All methods described hitherto used unweighted binary prediction values. Vocabularies based on different weight assignments were created using weights of $\alpha = 2$ (referred to as for example B380W2) and $\alpha = 5$.

In summary, the following binary prediction methods were applied:

(a) without using word frequency information, one *open* (OBO9901) and one *closed* (OBC380)

(b) including word frequency information, one *open* (BO9901) and three different *open* ones with time-windows of 190 days, 380 days and 570 days (B190, B380 and B570 respectively).

(c) using weights for the binary prediction value: BO9901W2, B190W2, B380W2, B380W5, B570W2.

All vocabulary selection methods based on word frequency information alone are visually depicted in Figure 9.5. Half a year of text data is represented by one block.

9.5.2 Results

In Table 9.2 on page 142 the mean OOV counts, the minimum and maximum values and the standard deviations of the different vocabulary selection methods are listed. In Figure 9.6 on page 143 the means and standard deviations are plotted. Below, the results are reported for every word selection strategy separately.

Static word frequencies

Of the vocabulary selection methods based on static word frequencies, as expected, the one that uses outdated data of 1999 (S99) has the highest overall number of OOV words. When the word selection procedure uses recent data, OOV performances improve, although the results indicate that the look-back time-window has to be sufficiently large. A look-back window of only half a year (S01L) shows a marginal but significant improvement in a paired t-test ($t = 3.1848, df = 33, p \leq 0.01$) of OOV performance compared to the method that uses outdated data, given it OOV rate of 1.94%. Increasing the window to a year (S01) or three years (S9901) results in a further improvement of the OOV performance to an OOV rate of 1,86% both. Given

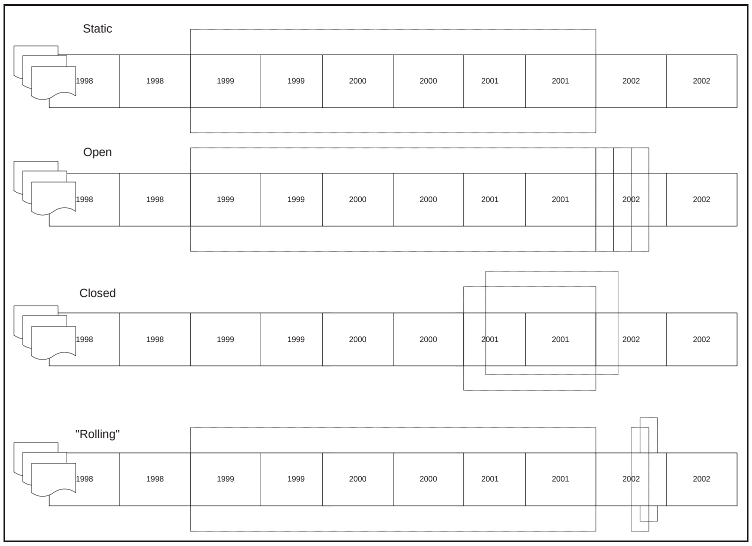

Figure 9.5: Vocabulary selection methods visualized. On top the static word frequency method, next the open word frequency method, followed by the closed word frequency method and the RLM method.

the comparable results of the last methods, looking back one year seems to be enough to make the best possible selection of words using static word frequencies. As the standard deviation for the longer time-window (S9901) is lower compared to the shorter time-window (S01), it seems that word selection becomes more robust using a larger time-window. However, this cannot be warranted as the variances do not differ significantly at the 5% level ($F = 0.9748$, $df = 33$).

In Figure 9.7 the OOV rates of the static word frequency methods are plotted for every week in the test data, with on the x-axis the testing weeks and on the y-axis the difference in OOV rate relative to the OOV rate of the 1999 baseline method (recall that for one week no OOV rates were computed as teletext data was missing completely for that week from the database). It shows that the method based on the last half year of 2001 (S01L) has sometimes even a worse OOV performance than the method based on outdated data.

Method	min	max.	mean OOV	mean OOV	stdev
			(cnts)	(rate)	
S99	123.00	806.00	485.91	2.06%	199.97
S01	106.00	796.00	440.50	1.86%	191.34
S01L	112.00	818.00	459.65	1.86%	196.61
S9901	106.00	786.00	440.38	1.94%	188.91
U9901	102.00	712.00	396.82	1.69%	162.71
CL168	108.00	719.00	404.21	1.72%	166.53
CL352	102.00	717.00	386.91	1.65%	160.91
CL520	110.00	734.00	393.79	1.68%	161.48
R01	106.00	710.00	399.71	1.70%	161.40
R005	106.00	700.00	392.59	1.67%	158.76
R0025	109.00	696.00	395.09	1.68%	159.49
OBO9901	120.00	842.00	486.26	2.07%	199.18
OBCL380	129.00	822.00	501.00	2.14%	200.64
BO9901	96.00	734.00	404.50	1.72%	169.47
BO9901W2	98.00	723.00	398.59	1.69%	164.49
B190	101.00	698.00	394.59	1.68%	161.22
B190W2	97.00	710.00	392.82	1.67%	162.26
B380	101.00	692.00	385.56	1.64%	158.77
B380W2	100.00	691.00	382.68	1.63%	158.14
B380W5	103.00	694.00	383.56	1.63%	157.67
B570	103.00	697.00	393.03	1.67%	160.48
B570W2	102.00	707.00	392.44	1.67%	160.45

Table 9.2: OOV performance statistics (minimum, maximum, mean OOV counts, mean OOV rate and standard deviation) of all word selection methods.

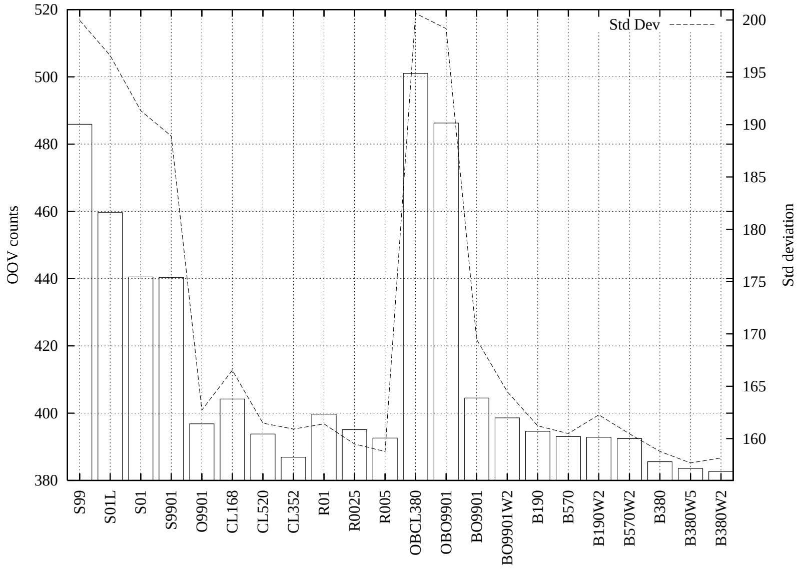

Figure 9.6: Mean OOV counts and standard deviations of all word selection methods.

Open word frequencies

Updating word frequencies while proceeding through time, as is done with the open word frequency method (O9901), has a relatively large positive effect on OOV performance. Compared to the best performing static word frequency methods (S9901 and S01), the mean OOV rate drops almost 10% to 1.69%. A paired samples t-test shows that the difference is highly significant ($t = 5.936, df = 33, p \leq .001$). To compare OOV rates on a weekly basis, in Figure 9.8 the OOV rate improvements on the 1999 baseline method of the open word frequency method and the merged results of the static word frequency 1999-2001 and 2001 methods, are plotted. The merging is done by taking the mean OOV rate for every data point.

The figure shows that the OOV rates of the open word frequency method improve relative to the static methods as the weeks in the experiment move further away from 2001. In the first weeks the open word frequency method performs only slightly better than the static methods, but the improvement increases slowly as the experiment proceeds. From the last week of April onward a sudden positive improvement shift can be observed. It must be noted that in this time period there was a lot of political consternation about a new Dutch political party and upcoming elections[4]. These events

[4] In May the leader of this party was assassinated, the next week this party had a very

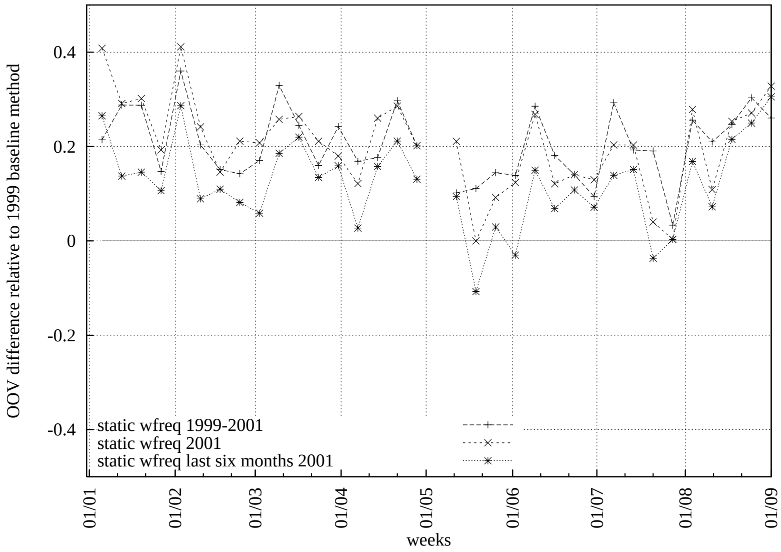

Figure 9.7: Improvements in OOV rates relative to the static 1999 baseline method (S99) of the static word frequency methods based on 1999-2001 (S9901), 2001 (S01) and the last six months of 2001 (S01L)

introduced a lot of words that were previously not seen very frequently. The open word frequency method apparently manages to capture some of these new words. Examples of frequent new words of the week of 12 May 2002 that are in the updated vocabulary and not in the static vocabulary are: "*LPF*", "*Herben*", "*Kamerzetel*", "*partijbaronnen*", "*Volkert*" and "*lijsttrekkersdebat*[5]": these are all words that do not, or do not very frequently appear in the complete 1999-2001 text data.

Closed word frequencies

Recall that the closed word frequency method was introduced to avoid that words that have built up a high frequency in the past, but should not be regarded as important words anymore, are included in vocabularies. The results of these methods, depicted in Figure 9.6, suggest that this procedure works, as long as the chosen time-window is neither too short nor too long. If the time-window is too short, represented by the CL168 method that uses a time-window of approximately half a year, OOV rates worsen a little relative to the open word frequency method (O9901): from an OOV rate of 1,69% to 1,72%. For larger time-windows, approximately a year (CL352) and one-and-a-half years (CL520), the closed word frequency method results in better OOV rates compared to the open word frequency method: 1,65% and 1,68% respectively. However, the difference between means is only significant for the closed (CL352) method ($t = 2.961, df = 33, p \leq .006$).

Note that the results of using only a time-window of half a year is in line with the results of the static word frequency method based on the last half year of 2001 (S01L): a time-window of only half a year is too short for an optimal word selection. Furthermore, the results indicate that a time-window that exceeds a year (520 days in the CL520 method) can undo some of the performance gain that is obtained by using a time-window of a year (352 days in the CL352 method): although small, the performance difference of these methods is significant according to the paired samples t-test ($t = 2.294, df = 33, p \leq .028$). Hence, for optimal word selection, a closed look-back window of one year seems the most appropriate word selection method hitherto. In Figure 9.9 the weekly OOV rates of the CL352 method and the open word frequency method (O9901) are plotted relative to the baseline method. Again, the sudden shift from the last week of April onward appears.

high score in the elections

[5] English translations: *LPF* – Lijst Pim Fortuin, a Dutch political party; *Herben* – name of a Dutch politician; *Kamerzetel*– Seat in the Dutch Lower Chamber; *partijbaronnen* – party barons; *Volkert* – first name of the man who assassinated a Dutch politician; *lijsttrekkersdebat* – party leader debate.

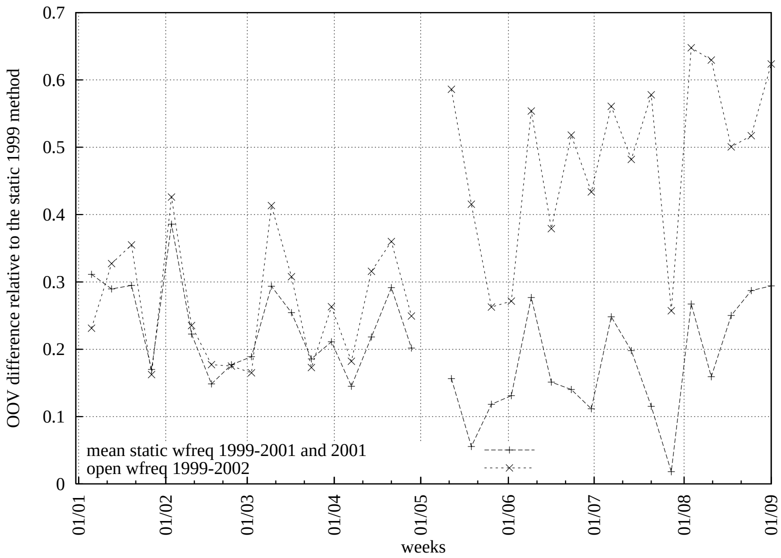

Figure 9.8: Improvements in OOV rates relative to the static 1999 baseline method of the merged static word frequency methods based on 1999-2001 (S9901) and 2001 (S01), and the open word frequency 1999-2002 method (O9901).

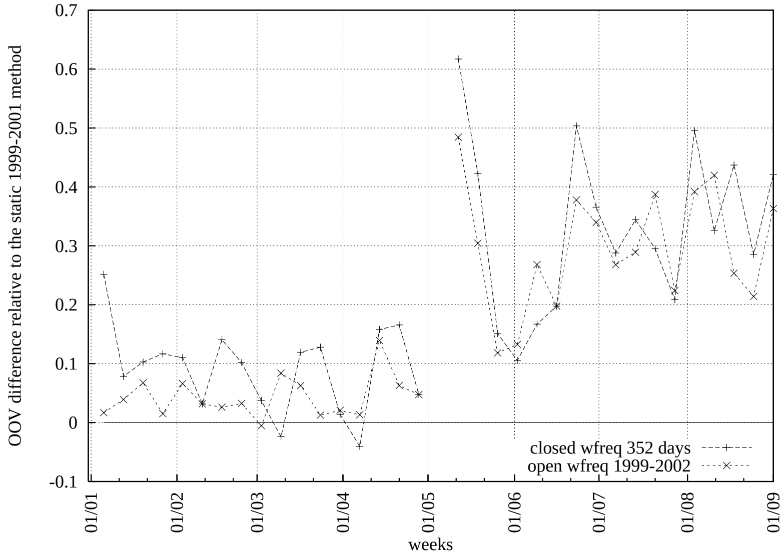

Figure 9.9: Improvements in OOV rates relative to the static 1999 baseline method of the closed word frequency method with a time-window of 352 days and the open word frequency method 1999-2002

Rolling language models

Unexpectedly, the word selection method based on the rolling language model strategy did not outperform the open word frequency method or the best performing closed word frequency method. Given that the RLM method is more complicated (using both updating and forgetting, definition of thresholds) relative to the plain updating of word frequencies, the result is somewhat disappointing. The RLM method with a relative word frequency threshold of 0.005 showed the best overall OOV performance, 1,67%, and does not differ significantly from the result obtained using the open word frequency method and the closed word frequency method based on 352 days.

binary prediction method

The vocabulary selection methods that are based on binary prediction values alone, hence temporal information only, evidently do not have a very high OOV performance as can be read off in Figure 9.6. However, when the time-window is long, as in the open binary prediction only method (OBO9901), the OOV performance is comparable with the static 1999 word frequency method (S99). Apparently there is too much loss of information when using binary prediction alone with a closed time-window of 380 days.

When word frequency information is added, OOV performance improves significantly. However, whereas the open binary prediction method (BO9901) outperforms the static 1999-2001 word frequency method (S9901), neither the weighted nor the unweighted variant reach the performance of the open 1999-2001 word frequency method (O9901). The weighted binary prediction method comes close but has still significantly more OOV words ($t = 5.7973, df = 33, p \leq .001$). The closed binary prediction methods do not show large improvements on earlier methods either. The closed binary prediction method with a 380 time-window (B380) differs marginally from the closed word frequency method with the 352 day time-window (CL352), although the difference is significant at the 1% level ($t = 3.6916, df = 33$).

When however the binary prediction values are weighted, the added value of the binary prediction method seems to grow. When a weight of $\alpha = 2$ is applied to the closed binary prediction method of a 380 day time-window, this method outperforms all word selection methods. A weight of $\alpha = 5$ shows a small degradation relative to $\alpha = 2$ but the difference is not significant ($t = 0,671, df = 33, p \leq .551$).

9.5.3 Discussion and conclusion

The experiment has shown that OOV performance of vocabularies benefits from an adequate temporal data selection procedure. OOV performance improves when more recent data is selected. A frequent adaptation of the vocabulary to changing word usage through time, provides additional OOV performance gain. In Figure 9.8 it can clearly be observed that OOV counts increase for vocabularies that are not adapted this way when the test data moves further away from the data selection period. Important news events that have such an impact on society that they dominate the news for a relatively long time, consolidate this effect and increase the necessity for adaptation of some kind. Updating word frequency counts using a parallel corpus such as newspaper text data and proceeding with the selection of the most frequent *N* words for the vocabulary, proved to be a simple but effective adaptation method. Best results are obtained when a closed look-back time-window of approximately one year is used. Providing that there is an up-to-date parallel corpus available, this method can be regarded as the most optimal solution for vocabulary selection, given its performance in proportion to its effort.

Parallel text source

It must however be noted that the reported results may change when other parallel text sources are used for adaptation. Up-to-date newspaper material is possibly not always directly available, forcing speech recognition system maintainers to use other, less extensive parallel sources for updating, such as teletext subtitling. The question is then, whether this may influence adaptation effectiveness. When the collateral data source is indeed teletext subtitling, adaptation effectiveness may be expected to decrease as newspapers will often have a small lead over broadcast news shows with respect to certain news items. The fact that in this experiment newspaper data was used as a parallel data source used for adaptation, may be an explanation for the disappointing results of the RLM method relative to the other methods. In Auzanne et al. (2000), compared to a static vocabulary selection method, OOV rates could be reduced with 20% using the RLM method, which is much larger than the improvements observed in this experiment. Auzanne et al. (2000) used news-wire text data for adaptation: it has a lower news coverage compared to newspapers. Therefore, it could well be that the disappointing performance of the RLM method is due to the relatively high performance of the other methods in the comparison, all based on newspaper data.

Window size

Concerning the window sizes that are used for updating word frequency counts, using more or less than approximately one year does not improve OOV performance. This was first observed with the static word frequency methods. Here the static 2001 method (S01) almost equals the performance of the static 1999-2001 method (9901). This tendency was also found with the closed word frequency methods, that gave the best OOV performance using 352 days, and with the binary prediction method that produced the best overall results with a time-window of well over a year. The deteriorating performance of a shorter time-window can be explained by the fact that such a time-window may not be able to capture words that have a not very high but at least a consistent frequency (see also the adaptations to the RLM method as discussed in Section 9.3), or words having a relatively high frequency that by coincidence were only sparsely observed in the time-window. The most probable cause of the performance degradation that is observed with larger window sizes, is that words that were important in a more distant past but are not relevant anymore in up-to-date news events are unjustly included in the vocabularies. Apparently, this already happens when the time window is prolonged to one-and-a-half years. This hypothesis is confirmed by the results of the closed word frequency methods compared with the results of the open word frequency method (O9901). The closed word frequency method was created especially to prevent that formerly important words are included in the vocabularies. Using the long time window (CL520), OOV performance improves slightly, but the best results are obtained using a window of a year (CL352).

Binary prediction

The differences in OOV performances of the binary prediction procedures in general justify a more detailed discussion. Starting with the procedures that only used temporal information for word selection, the results show that long-term temporal information (OBO9901) bares more predictive information than short term temporal information (OBCL380). The long-term method has an OOV performance comparable with a method based on outdated static word frequency information, which is not really high, but at least indicates that temporal information has a reasonable predictive power for word selection. When temporal information is combined with word frequency information, OOV performances improve drastically, but the question is, what exactly the separate information sources contribute to this improvement. Comparing the open word frequency method (O9901) with the open binary prediction method (BO9901, BO9901W2) –so, comparing long-term, updated word frequency alone with a combination of long-term updated word frequency information and long-term updated temporal information– a small but significant performance *degradation* is observed when temporal information is included. However, when including and excluding temporal information is compared for relatively short-term word frequency and temporal information (Cl168 with B190, CL352 with B380, CL520 with B570) a reverse tendency can be observed. Methods that also incorporate temporal information have a slightly better performance than methods that use word frequency information only. This seems to contradict earlier findings that long-term temporal information contains more predictive information than short term temporal information. However, one can argue that such a conclusion cannot honestly be drawn as in these earlier findings, single information sources were compared whereas the latter comparisons include different information sources. It is possible that the opposite effects on OOV performance of adding temporal information, can be explained by the performance of word frequency information itself. When word frequency information is not robust enough, for example when long-term word frequency information is used (O9901), adding temporal information only worsens word selection performance. When however word selection has already reached a good OOV performance on the basis of word frequency alone, temporal information can improve word selection. Although such an explanation fits in the observed results, the reported evidence is evidently too small to warrant any strong conclusion in this direction.

Final conclusions

On the basis of the results the following final conclusions can be made:

- A selection of approximately one year of recent newspaper data, may be regarded as an optimal starting point for a word frequency based selection of vocabulary words for the representation of Dutch news broadcasts. The results of the experiment suggest that one year is long enough to capture words that have a not very high but consistent frequency, and short enough to reduce the chance of unintentionally including words in the vocabulary that have built up high frequency counts in the past but are not relevant anymore in up-to-date news events.

- In longitudinal speech recognition tasks in the broadcast news domain, periodic updating of the vocabulary is necessary as to enable the recognition of new words that gradually appear over time.

- A simple but effective procedure for periodic updating is using a shifting look-back time-window of approximately one year for word selection based on word frequency counts.

- The results of the experiment suggest that including temporal word usage information in the word selection procedure can improve OOV performance of vocabularies additionally provided that the word frequency information is already robust in itself.

9.6 Summary and final remarks

In this chapter, the selection of words for the language model vocabulary was addressed. It was argued that an optimal word selection procedure is important as to reduce the number of out-of-vocabulary words in speech recognition tasks. Word selection was addressed as being a matter of appropriate training data partitioning: either on the basis of content information or on the basis of temporal information. Using temporal information for data partitioning was discussed in more detail. A number of vocabulary selection methods based on temporal data partitioning and word frequency information were discussed. It was argued that word frequency information alone is inadequate to deal with word importance fluctuations over time. In order to capture such dynamics in a domain that typically shows large word importance fluctuations, the broadcast news domain, a new method was introduced, called the binary prediction method. This method tries to incorporate temporal information directly into the selection procedure. Indeed, this method gave the best OOV performance in a vocabulary selection experiment, that compared a number of different vocabulary selection techniques. However, the gain was very small and it may be worthwhile to address more research to the implementation of the binary prediction method. For complexity reasons it was chosen to use temporal information in a highly simplified way and in a compressed format. It can be argued that by doing so, only little temporal information remained to be helpful in the vocabulary selection procedure. A improved paradigm for representing temporal information may result in a more substantial OOV performance improvement.

Chapter 10

Compound splitting

This chapter addresses the splitting of compound words in order to improve large vocabulary speech recognition performance. A data-driven compound splitting algorithm is described and language model performances are compared in terms of out-of-vocabulary rates and word error rates.

10.1 Introduction

In the previous chapter the phenomenon of compounding in Dutch was discussed in the context of lexical coverage: as words can be joined together almost freely to form new words the number of distinct words in Dutch is relatively large (even theoretically infinite) compared to non-compounding languages such as English or Italian. Take for example, the valve cap of a bicycle tire that can be translated into a single compound word in Dutch: "*fietsventieldopje*". Likewise, Dutch has the compound "*autobandventieldopje*" which translates to the valve cap of a tire of a car, and so on. Given this agglutinative behaviour, it was concluded that for a compounding language the vocabulary space is extra-sparse and that it is therefore important to select accurately the words that are expected to appear in the task domain. Different vocabulary selection methods, as discussed in the previous chapter, can be applied for this purpose, but another, or possibly additional approach, would be splitting or decompounding compounds that are encountered in the language model training corpus, back into pairs (or n-tuples) of single constituents. The compound "*autobandventieldopje*" could for example be decomposed into "*auto band ventiel dopje*". This procedure evidently increases the total number of words, but will usually reduce the number of distinct words as single constituent parts are also expected to appear on their own and in other compounds. From a lexical coverage perspective, a reduction of distinct words in a corpus representing a certain task domain, means that with the same vocabulary size, more words can be covered in the domain, or the inverse, less words will be out-of-vocabulary (OOV). The importance of reducing OOV words for a speech recognition system's vocabulary was discussed earlier. One OOV word can result in more than just one recognition error, as the word that replaces the OOV word in the recognition process, often damages the n-gram prediction of the next word. Adda-Decker and Adda (2000) investigated compound splitting for German in order to limit lexical variation in text corpora. By splitting compounds in the corpus, OOV rates of $65\,K$ lexicons could be reduced by almost 20%. Preliminary studies (Ordelman et al., 2001b,c,d; Ordelman and De Jong, 2003) indicated that compound splitting can be beneficial for Dutch lexicons as well.

But although compound splitting may improve lexical coverage and reduce OOV words for speech recognition vocabularies, it is uncertain whether it improves overall speech recognition *performance*, as has often been suggested but never adequately investigated for Dutch. There are a number of side-effects of compound splitting that may undo a possible speech recognition performance gain due to an improved OOV rate. Such disturbing side-effects can be classified according to the different stages in the recognition (development) process in which they occur:

- Acoustic modelling

  From an acoustic modelling point of view it is easier to recognise longer words than shorter words as longer words bear more acoustic information. Some evidence for the reduced speech recognition accuracy caused by the introduction of short compound constituents was found in Berton et al. (1996).

- Dictionary generation

  The phonetic transcriptions of former compound parts may depart from the actual pronunciation of the compound when co-articulation effects occurred at constituent boundaries. The transcription of a compound reflects existing within-word co-articulation effects, such as the unvoiced *[* t *]* that changes into a voiced *[* d *]* before a voiced plosive as in "*voetbal* English: football" that is phonetically transcribed as *[* v u d b ɑ l *]*. Supposing that the word "*voetbal*" is decomposed into the words "*voet*" and "*bal*" the co-articulation effect disappears in the tuple of compound parts as the final *[* t *]* in "*voet*" is pronounced as *[* t *]*. Consequently, there will be a mismatch between the actual pronunciation of a compound and the phonetic representation in the phonetic dictionary of the recogniser.

- Language modelling

  It cannot directly be foreseen what the effect of compound splitting is on n-gram estimation. The n-gram information is practically reduced to the $(n-1)$-gram information as the decomposed compound pushes one or more context words out of the n-gram. For example, suppose that a compound "*ventieldopje*" was well modelled using a standard

> text corpus. After compound splitting, the frequent trigram "*ventiel op de*" may disturb the prediction of "*ventiel dopje*".

The question is to what extent the positive effect on speech recognition performance of an improved lexical coverage will be reduced by the spreading of the acoustic information over word-tuples, possible mismatches in phonetic representations of compound parts and the loss of context information for the n-grams. One could anticipate some of these side-effects by applying restrictions to the compound splitting procedure aiming at speech recognition performance optimisation, for example, by setting a minimum word length for the compound parts, or by restricting compound splitting to low frequent compounds. Highly frequent compounds normally have a high chance of being recognised correctly. Their length guarantees a relatively large amount of acoustic information that can be used for its recognition. In addition, as their frequency is high, the n-gram estimates may be expected to be reasonably well trained. One can argue that is therefore preferable not to decompose such compounds, in spite of the fact that it would improve lexical coverage. This approach can be related to methods that try to reach at speech recognition performance improvement by combining frequent orthographic word tuples, referred to as multi-words, into single items in the recognition lexicon (Gauvain et al., 1997) instead of decomposing words.

Restricting the compound splitting procedure can also be based on other grounds. From a lexical coverage optimisation point of view, it can be argued that it may not always be beneficial to decompose *every* compound. Every decomposition of a compound involves a re-ranking of words in a frequency-sorted word list. Given that the vocabulary is selected out of the N most frequent words, words will migrate from the out-of-vocabulary space to the vocabulary space and vice versa due to the compound splitting procedure. When a highly frequent compound is split into two highly *infrequent* parts that by themselves would not be incorporated in the vocabulary, the decomposition has only little added value. Instead, as the infrequent parts at once become very frequent, and as a consequence, enter the vocabulary space, another frequent word has to be removed from the vocabulary space to keep an equal number of words. The final effect of compound splitting may therefore be reduced coverage. This type of restricted compound splitting, aiming at a better lexical coverage, will be discussed in detail in Section 10.4.

A complicated factor regarding compound splitting in a Dutch speech recognition context, is the handling of the binding morpheme "s". For Dutch, it is allowable to insert this binding morpheme between specific constituents[1], as in "*regering-s-leider* (English: leader of the government)". There are three possible approaches for dealing with the binding morpheme in compound splitting:

[1] Whether the insertion of a binding morpheme is correct or incorrect is formalised in "*Het Groene Boekje*" containing Dutch spelling rules

- Interpret the binding morpheme as a single constituent

  In this approach the binding morpheme becomes a lexical unit in itself (*regering s leider*), ignoring the fact that in this way units are introduced that are not linguistically meaningful (see the discussion about linguistically meaningful units below). This benefits the reduction of distinct words as the surrounding words will not be "contaminated" by the binding morpheme so that introduction of new lexical units (e.g., *regerings* which is not a regular word in Dutch) can be prevented. The disadvantage of this approach is that the mono-phone binding morpheme is vulnerable to recognition errors.

- Attach the binding morpheme to the preceding word

  This approach (*regerings leider*) will prevent the introduction of recognition errors originating from a stand-alone binding morpheme. The disadvantage of this approach however, is that new lexical units are introduced so that the reduction of distinct words, the main goal of compound splitting, will be smaller.

- Remove the binding morpheme

  Crudely deleting the binding morpheme (*regering leider*) removes possible disturbing effects of the binding morpheme mentioned in the two approaches above: the binding morpheme cannot be incorrectly recognised and the reduction of distinct words is maximised. However, this approach at least introduces another error source as existing acoustic evidence (the "s" that is pronounced) cannot be accounted for anymore.

Finally, the question must be addressed as to whether the application in speech recognition, a compound splitting algorithm needs to produce linguistically meaningful units. It does not, some would say, as long as the number of correctly recognised words grows after the reconstruction of a stream of either meaningful of meaningless units. It must be noted however that it is exactly the reconstruction process that may be difficult. As some units will be miss-recognised the reconstruction process can either be impossible (there are no valid combinations) or produce an incorrect reconstruction (the combination is valid, but the reconstructed word is not the original word, only a part of the word was recognised correctly). Whether this is really a problem depends on the task for which speech recognition is deployed. At least in an information retrieval framework, one could easily skip the reconstruction step and apply the same compound splitting procedure to all words encountered in the process (query, parallel corpora): retrieving the right documents is the primary aim, not the linguistic validity of the speech recognition transcripts. Having recognised any part of it with this respect, is always better then not having recognised it at all. But also for other tasks, the benefits (assuming they exist) of compound splitting may hold, despite incorrect reconstructions. Supposing that originally a compound could not be recognised correctly, an incorrect reconstruction after a speech recognition run that uses compound splitting, will not worsen the transcript. In the worst case, the compound is still not recognised correctly, but possibly at least a part of it is correct. In this research however, the approach aimed at a decomposition into existing words, hence linguistically meaningful units, as this also guarantees a high coverage of the phonetic dictionary that is used. Units that are not linguistically meaningful will evidently not exist in standard phonetic dictionaries. Moreover, such units may have infrequent phone combinations at the constituent boundaries which may damage transcription accuracy of grapheme-to-phoneme transcription tools.

To investigate the effect of compound splitting with and without restrictions on speech recognition performance, language models were created based on newspaper text data. In the next section (Section 10.2), the splitting algorithm to create the decomposed text versions is described, followed by an evaluation of unrestricted compound splitting in terms of lexical coverage (Section 10.2.4). Next, restricted compound splitting aiming at improving lexical coverage is discussed in detail (Section 10.4). In Section 10.5, the language models created using unrestricted and restricted compound splitting methods are evaluated in a broadcast news transcription task. The results of the evaluation are discussed in Section 10.5.3.

10.2 Splitting algorithm

10.2.1 Introduction

Detecting compound words accurately is difficult and actually requires a refined morphological analysis. In some cases, also semantic information is needed to decide on the validity of a decomposition. As morphological and semantic analysis tools were not available for this research, a compound splitting method had to be found that does not require higher level information sources and has a performance that enables a proper investigation of the effect of compound splitting on speech recognition.

A number of compound splitting methods are discussed in the literature. In Adda-Decker and Adda (2000), a limited set of 335 German decomposition rules were developed empirically using a newspaper corpus. In this study, the decomposition of compounds was investigated related to lexical coverage in speech recognition and grapheme-to-phoneme translation. In Pohlmann and Kraaij (1996), Dutch compounds are decomposed based upon a compound well-formedness table of allowed syntactic class combinations, proposed by Vosse (1994). This method requires labelling of compound parts with syntactic categories so a lexicon with syntactic information is needed. In this study, compound splitting was investigated in an information retrieval framework. Also for the use in information retrieval, Monz and de Rijke (2002) implemented a character-based noun-noun compound splitting algorithm. This algorithm tries to split an input string at every character position until a noun is identified as the prefix of the string and the remaining part can be (recursively) identified as a noun as well. For the identification of the nouns the Dutch CELEX lexicon (Baayen et al., 1993) was used. A data-driven compound splitting method, applied in a German speech recognition task, was proposed by Larson et al. (2000). Compound words are split according to the statistical relevance of iteratively generated splitting points. In this study the frequency of words containing a potential constituent were used to define local maxima. When, moving both from left-to-right and from right-to-left through the compound, at some point a local change maximum was encountered, this splitting point was regarded as relevant.

For the development of the grapheme-to-phoneme converter (see also Chapter 4), a large pronunciation lexicon (large part of "De Grote Van Dale" Dutch dictionary, further referred to as GVD lexicon) comparable with the Celex lexicon[2], was used. This lexicon could be used in a compound splitting algorithm as developed by Monz and de Rijke (2002). However, it was decided not to use this method and the GVD lexicon for a number of reasons. Although the GVD lexicon has a relatively high overall coverage of the words encountered in the newspaper corpora, a preliminary study showed that compound coverage was relatively low. Compound words are "invented" every day and as the GVD lexicon is a few years old, recently invented compounds will be missed. Moreover, the policy of dictionary publishers usually is not to include all possible compound words in dictionaries. Most of the times, a few frequent examples within a semantic context will do: when "*aarbeienjam* (English: strawberry jam)" en "*bosbessenjam* (English: blueberry jam)" are given, another type of preserve such as "*perenjam* (English: pear jam)" is regarded as redundant. Furthermore, the GVD lexicon contains many items that are not words. As the lexicon was compiled for the training of grapheme-to-phoneme conversion tools, non-word lexical items such as "achtig (English: ish)" and "elijk (English: ly)" were included. These occurrences could severally damage compound splitting performance.

To enable compound splitting of recent compound words, a data-driven method seemed the best option. However, the method of Larson et al. (2000) is computationally expensive compared to an alternative data-driven approach that is reported here. Instead of using the GVD lexicon, the word list of all words in the newspaper corpus was used as a starting point for compound splitting. At least, this list would also contain recently invented compounds. However, as this word list contains many non-word lexical items as well, the compound splitting algorithm had to adapted to reduce the number of false alarms or incorrect compound splittings due to the occurrence of such items. A "greedy" compound-search algorithm was de-

[2] The GVD lexicon does not however include syntactic category labels

veloped that uses sorting, word length information and word frequency information to detect and split compounds. It must be noted that it was not the intention to develop a perfect compound splitting algorithm, but instead an algorithm that has a compound splitting performance with a high recall and precision score so that the effect of splitting compound words on speech recognition performance could be adequately investigated.

10.2.2 Search algorithm

First, a compound was defined as a word that can be split into at least two separate words, an α and a β constituent, that both exist as single words with a minimum frequency of 10 in the text database. The minimum frequency was introduced to avoid that words that normally do not occur in Dutch as single items but by accident[3] appear in the text data, produce incorrect compounds. Furthermore, both constituents of a compound must have a minimum length of six characters. This restrictions was imposed to the splitting procedure to reduce the amount of false compound detections, such as in *voorstel* "English: proposition" that should not be split into "*voor* (English: before)" and "stel (English: pair)". Note, that this restriction also reduces possible disturbing effects on speech recognition performance in advance: the number of words that are six characters and shorter —words that are in principle harder to recognise correctly due to acoustic confusability as discussed earlier— will not increase by the compound splitting procedure. Finally, a compound was allowed to have a binding morpheme "s" that at this stage was interpreted as a stand-alone constituent (e.g., *regering-s-leider*). The decision to attach the binding morpheme to the preceding word or delete it completely, as suggested as possible approaches for dealing with the binding morpheme discussed in the introduction, was postponed to later processing stages.

To collect the largest possible number of compounds, a word list of more then 1,5M unique words collected from the available text data, was alphabetically sorted. In this way, the first part of a compound, further referred to as the α-constituent, always precedes the compound: "*voetbal* (English: football)" for instance, precedes compounds such as "*voetbalschoen* (English: football boot)", "*voetbalstadion* (English: football stadium)" and so on. By descending the word list and checking if the current word is used as an α-constituent in the next entries and the remainder of the word exists as a single word in the list as well, compound words could be detected. Note that words with an initial uppercase were discarded to avoid false compound detections such as "*Barend-recht*", which is the name of a city that should not be decomposed into a name ("Barend") and a word ("right")[4]. In order to find possible alternative splitting methods for a words, this method was repeated using a list of words in reversed order

[3] Due to normalisation procedures or because they are of foreign origin

[4] An adequate normalisation of the text data is important

so that words became search key for final constituents: the word “stadion (reverse: noidats)” could for instance be found as final constituent in “voetbalstadion (reverse: noidatslabteov)”. A third compound-search detected words with constituents ending in the suffixes “*ing(s)*”,”*ingen*”, “*heid(s)*”, “*heden*”, “*schappen*”, “*schap(s)*”. When these suffixes appear in the middle of words, these words are always compounds that can be split after this suffix. The greedy compound search algorithm found 323.213 compounds with at least two constituents in the first run. These compounds were put in a conversion table with the compound in one column and a compound splitting solution in the other.

10.2.3 Multiple splitting alternatives

Of 6052 compounds the algorithm produced two or more possible splitting solutions. This happened for example when a compound could be split into three or more constituents, such as in “*wassenbeeldengallerij* (English: waxwork gallery)” that can be split into “*wassenbeelden-gallerij*” (preferred) and “*wassen-beeldengallerij*” (semantically poor). This is usually not problematic however, as the compound that is left after a first decomposition step, is decomposed in successive steps of the iterative compound splitting procedure. However, other examples, such as those in Table 10.1 also produce implausible compound splitting alternatives[5]. Although a majority of the decomposition alternatives produced by the compound search algorithm are *possible*, not every alternative is *plausible*. A native speaker of Dutch will in general be able to pick the most plausible one out of multiple compound splitting alternatives although in some cases context information is required to make the decision. In order to select the most plausible compound splitting solution automatically, information had to be found that could help to make this decision automatically.

compound	plausible	implausible
reactiestappen	*reactie-stappen*	*reacties-tappen*
koningspaarden	*koning-s-paarden*	*koning-spaarden*
meubelsmokkel	*meubel-smokkel*	*meubels-mokkel*
schijntrappen	*schijn-trappen*	*schijnt-rappen*
politiekringen	*politie-kringen*	*politiek-ringen*

Table 10.1: Compounds with more than one possible compound splitting

[5] Translations of the compounds in Table 10.1 from above: *reaction steps, horses of the king, smuggling of furniture, dummy kicks and police circles*

Observing the list of multiple splitting alternatives, it was noted that often the most plausible alternatives were constructed out of constituents that were in general words with a higher frequency than the ones in the implausible alternatives. Therefore, it was assumed that general word frequency could be deployed for deciding which of the alternatives is the most plausible one. Supposing the task is to fill two empty compound slots with words, so that the concatenation of the two slots produces a compound word C, the probability that a particular word α is chosen for the first slot out of all possible words in our corpus is its relative frequency in the corpus, defined as:

$$\frac{f(\alpha)}{N} \tag{10.1}$$

Assuming that the second slot is filled independently from the first slot[6], the probability of filling both slots with two given words α and β is the product of the two relative frequencies:

$$P_1(\alpha\beta|C) = \frac{f(\alpha)}{N} \cdot \frac{f(\beta)}{N} \tag{10.2}$$

A first test run showed that using this information source alone could not provide enough information for a successful detection of the correct compound splitting alternative. Therefore, a second observation in the list of multiple splitting alternatives was deployed in the decision procedure: there are a large number of compounds that share the first constituent, for example "*drugs*". The probability of "drugs" being the first constituent of a compound may therefore be regarded as relatively high. In the opposite case, for other words it is not likely that they appear as first or final constituent of a compound. By computing the *within-compound frequency* of constituents this information could be used to detect the most plausible alternatives. This within-compound constituent probability was computed by counting all occurrences of a particular constituent α in the set of compounds ($f_C(\alpha)$) and normalising over all compounds (N_{comp}):

$$\frac{f_C(\alpha)}{N_{comp}} \tag{10.3}$$

The probability that two particular constituents α and β make a compound C was then defined as:

[6] This is in practice obviously not true as the concatenation of words will be forced by semantic considerations.

$$P_2(\alpha\beta|C) = \frac{f_C(\alpha)}{N_{comp}} \cdot \frac{f_C(\beta)}{N_{comp}} \tag{10.4}$$

Finally, the general constituent frequency and within-compound frequency information were combined in a plausibility measure $\mathcal{Q}_{split}$, a multiplication of the probability estimates of both sources :

$$\mathcal{Q}_{split} = \overbrace{\left(\frac{f(\alpha)}{N_{const}} \cdot \frac{f(\beta)}{N_{const}}\right)}^{overall} \cdot \overbrace{\left(\frac{f_C(\alpha)}{N_{comp}} \cdot \frac{f_C(\beta)}{N_{comp}}\right)}^{within-compound} \tag{10.5}$$

To check the performance of the decompound probability measure, the GVD lexicon described above was used as a reference. For every compound that had splitting alternatives, it was checked if the compound appeared in the GVD lexicon. If so, the splitting alternative suggested by the algorithm (Equation 10.5) was compared with the compound splitting solution suggested by the GVD lexicon. Of the 6052 compounds with multiple alternatives, 708 existed in the GVD lexicon as compounds, 13 compounds appeared to be false alarms as these existed in the GVD lexicon but were not regarded as compounds. Of the 721 compounds that were checked, 621 of the decompound solutions corresponded with the solution provided by the GVD lexicon, 87 solutions were different (see also Table 10.2). Given this score, the detection of the most plausible compound splitting alternative was regarded as reasonably successful. The compound conversion table discussed so far only provided a pairwise compound splitting solution with only an α and a β constituent. A compound, such as for example:

	compounds with alternatives	percentage
total	6052	
in GVD	721	11.91%
not in GVD	5331	88.09%
correct	621	86.13%
with errors	100	13.87%
false alarms	13	
incorrect	87	

Table 10.2: Alternative selection evaluation statistics.

"*wassen-beelden-gallerij*",

that should be split into three constituents, was given the compound splitting:

"*wassenbeelden-gallerij*"

or:

"*wassen-beeldengallerij*".

These compounds had to be decomposed in repeating runs until no more compounds were detected in the decompound column of the conversion table. In the next two runs that were needed, a total of 12740 entries were altered this way.

10.2.4 Evaluation of splitting accuracy

To evaluate the complete compound splitting table, the same procedure as with the alternative detection evaluation was applied. If a compound appeared in the GVD lexicon the compound splitting solutions produced by the splitting algorithm and those provided by the GVD lexicon were compared. If the GVD lexicon did not provide a compound splitting solution, it was assumed that the algorithm had produced a false alarm: the compound apparently was not a compound. Furthermore, as the intention was to develop a compound splitting table with an optimal performance, the performance information was directly used to improve the compound splitting table. So, after the evaluation of the initial compound splitting step, detected incorrect splitting solutions were replaced by the correct solutions from the GVD lexicon. The corrected table was then used in the steps that followed.

It must be noted that this way only a biased subset of the compounds could be checked, namely those compounds that also appeared in the GVD lexicon. Going manually through the other compounds however did not reveal unexpected compound or splitting occurrences. Unfortunately, because of the large data set, the recall of the compound splitting algorithm could not be tested but due to the design of the algorithm it is expected to be high: only compounds that contain a word that was not seen as a single lexical item in almost four years of newspaper data could not be detected.

In Table 10.3 the scores of the compound splitting steps are listed. First, the performance of the initial compound splitting step was evaluated (first iteration). As this step produced compound splitting solutions with two constituents only, compound splitting solutions with more than two constituents in the GVD lexicon could not directly be compared. Therefore, a compound splitting solution was regarded to be correct if its boundary matched one of the compound boundaries given in the GVD lexicon. The precision score after the initial compound splitting step is 96.69 %. After the decstep1 evaluation 113 incorrect splitting solutions were corrected in the compound splitting table. The 764 false alarms and GVD corrections with constituents smaller than 6 characters were deleted (1237 in total). With the corrected compound splitting table the splitting solutions were decomposed in two subsequent steps. In the second iteration 10,057 entries were altered (2-tuples to at least 3-tuples), in the third the last 43 entries. As detected incorrect compound splittings were corrected after every step, no errors could be detected after the second iteration. Given the high percentage of correct splittings (precision) after the first splitting step, the final conversion table is expected to have a high precision score as well. With this table, newspaper text data can appropriately be decomposed so that the effect on speech recognition performance can be investigated. But to ensure that lexical coverages improve by splitting compound words with the described splitting method, first the lexical coverages of vocabularies derived from the original and decomposed text data were compared.

after 1 iteration	entries	323,213	
	in GVD	40,806	(12.63 %)
	not in GVD	282,407	(87.37 %)
	correct	39,457	(96.69 %)
	false alarms	764	
	incorrect	585	
	with errors	1349	(3.31 %)
	deleted	1237	
	corrected	113	
	entries remaining	321,977	
after 2 iterations	further decomposed	10,057	
	total in GVD	39,570	
	total not in GVD	282,407	
	correct	39,562	(99.98 %)
	incorrect	8	
	deleted	2	
	corrected	6	
	entries remaining	321,975	
after 3 iterations	further decomposed	43	
	in GVD	39,568	
	not in GVD	282,407	
	correct	39,568	(100 %)
	incorrect	0	
	entries remaining	321,975	

Table 10.3: Evaluation statistics of three splitting iterations. Only the splittings of compounds were checked that appeared in the GVD lexicon.

10.3 Lexical coverage evaluation

In order to evaluate whether compound splitting improves lexical coverage, a relatively large text collection of well over 300 *M* words of newspaper data from January 1999 until December 2001 was used. From this collection word frequency files were created, one based on the original text version and one based on a version that was decomposed without applying any restrictions. The word frequency file of the original text data contained 1,322 *K* words, the one of the decomposed data 1,050 *K* words. Hence, compound splitting resulted in a 20 % reduction in distinct words. Lexical coverages of vocabularies, created by taking the top *N* words of both word frequency files, were computed cumulatively, starting with taking only the first word in the word frequency file ($N = 1$) and ending with taking all words in the word frequency file ($N > 1\,M$). In this way, lexical coverage statistics became available for every vocabulary size, so that improvements (or deteriorations) in lexical coverage due to compound splitting could be compared for any given lexicon size. Figure 10.1 shows the differences in percentage lexical coverage of lexicons based on decomposed data relative to the ones based on the original data. Only lexicon sizes up to 100 *K* words are shown. The figure shows that compound splitting has a negative effect on lexical coverage of very small vocabularies, a large positive effect on lexical coverage up to a vocabulary size up to some 20 *K* words and this positive effect slowly decreases when larger vocabularies are used. After 100 *K*, not shown in this figure, lexical coverages after compound splitting remain higher but the differences decrease. In general, this result can be interpreted as a confirmation of the hypothesis that lexical coverage of a vocabulary improves when compound words are decomposed, with an exception for very small vocabularies. An explanation for this and for the slow decrease in improvement with vocabularies larger then 20 *K* word, can be found by looking more closely at the lexical coverage function:

$$LC(w) = \frac{\sum_{w=1}^{L_{size}} f(w)}{\sum_{w=1}^{N} f(w)} = \frac{L}{N} \tag{10.6}$$

where $f(w)$ is the word frequency of word w in a development corpus, L the lexicon, and N the total number of words. Firstly, although compound splitting reduces the number of distinct words, it increases the total number of words as every decomposed compound word, is substituted for at least two other words. The denominator of Equation 10.6, referred to as total word mass (TM), will therefore be higher after compound splitting. For very small vocabularies, the increased total word mass has a negative effect on lexical coverage. With a growing vocabulary, it takes some time to overcome this negative effect: a break-even is reached (at a vocabulary size of 1415 words) at the point where the vocabulary has gained enough

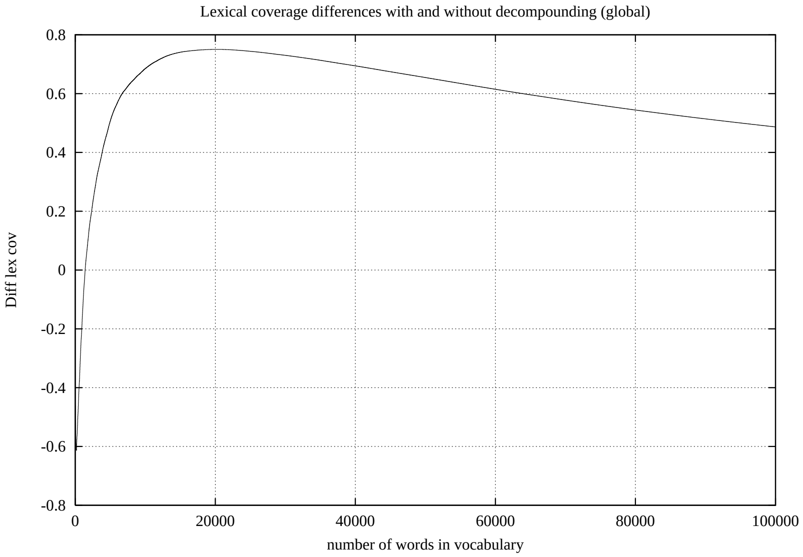

Figure 10.1: Differences in percentage lexical coverage (y-axis) before and after compound splitting as a function of vocabulary size (x-axis).

word frequency mass by incorporating words that are parts of frequent compounds to undo the initial coverage loss. From that point onward, lexical coverage improves as more and more compound words that could originally not be covered have their parts contained in the vocabulary. A maximum lexical coverage improvement is reached at a vocabulary size of 20414 words: lexical coverage after compound splitting is at that point 0.75 % higher (absolute) then before. With larger vocabularies, the lexical coverage improvement slowly drops. Evidently, the effect on lexical coverage of adding one word to the vocabulary becomes smaller as the word frequency list is descended: the frequencies of these words are lower so only a few extra words are covered afterward. On top of that, the effect of compound splitting also decreases as compounds have a lower word frequency and less frequent constituents. Suppose that such a compound was not decomposed, missing it in the vocabulary would have only a small effect on lexical coverage anyway. With an increased total word mass, splitting such compounds will subdue a lexical coverage improvement.

But as it is uncertain whether other dynamics have had a part in the decrease in lexical coverage improvements, it seemed worthwhile to investigate if compound splitting must be restricted to reach even higher lexical coverages. In the next section possible compound splitting scenarios are therefore looked at more closely.

10.4 Restricted compound splitting

Although compound splitting improves lexical coverage of lexicons larger than some 1500 words, it may be advantageous *not* to decompound every compound word that is encountered. The results described in the previous section suggests this and looking more closely at possible compound splitting scenarios seems to confirm this hypothesis. In figure 10.2 these (bipartite) compound splitting scenarios are shown. Both figures represent a word list, the top area being the lexicon of a given size. The area below the lexicon boundary (including the “buffer” which is the lexicon size plus one) contains the words that are out-of-vocabulary. The figure on the left represents those situations in which the compound word was in the lexicon (first global scenario). The figure on the right shows those situations in which they are not (second global scenario). The arrows represent migrations of constituents (words) in and out of the lexicon. For every global scenario three local scenarios can be distinguished (marked A, B and C). Figure 10.4 illustrates the actual distribution of compounds in the data. It shows the number of compounds per $10\,K$ words in the numerically sorted word frequency list. The largest number of compounds can clearly be found outside the range of lexicons of realistic size.

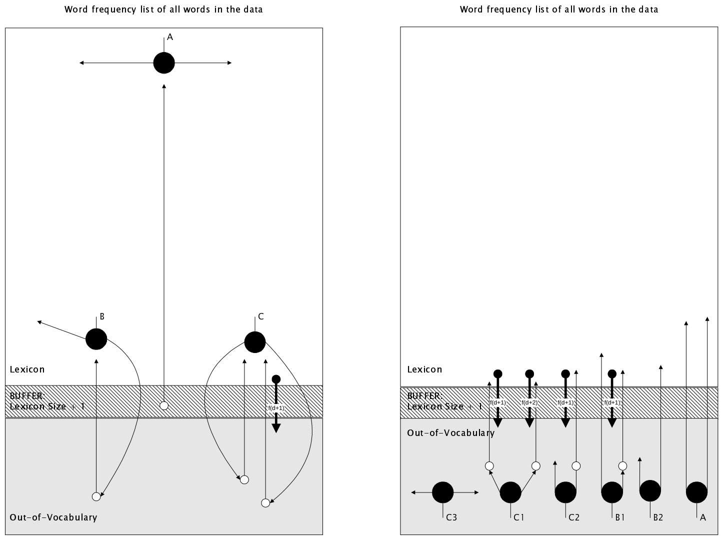

Figure 10.2: Constituent migration of compounds that are in the lexicon (left) or not in the lexicon (right). For every figure, three global scenarios can be distinguished (marked A, B and C). An arrow going from a point represents a constituent that migrates to a certain position in the word list. The black arrows represent a word that is thrown out of the lexicon.

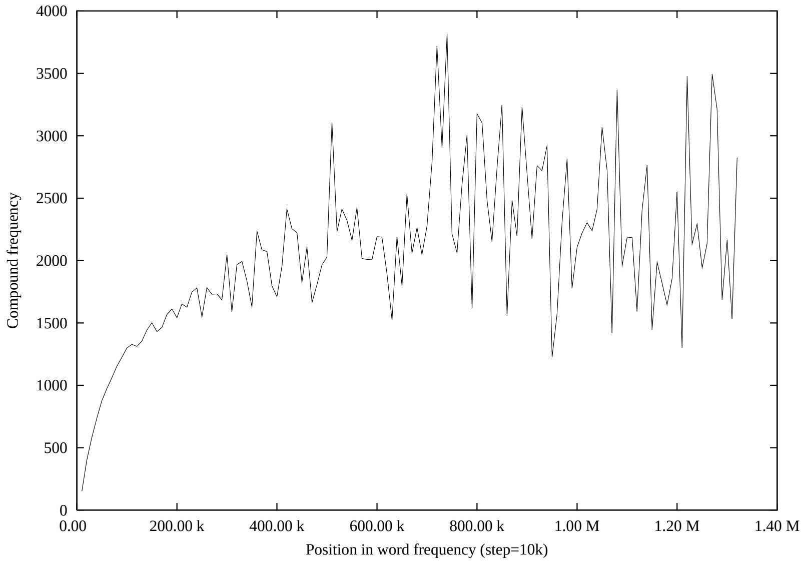

Figure 10.3: Frequency of compounds in word frequency file. Every data point represents the number of compounds in 10k words of the word frequency file

To compute the exact effect on lexical coverage of splitting a compound in one of the scenarios depicted in Figure 10.2, every scenario was modelled using the lexical coverage function (Equation 10.6). After compound splitting an individual compound C, lexical coverage should be higher then before compound splitting it. In other words, the ratio of the lexicon mass after compound splitting (LM_{new}) and total word mass after compound splitting (TM_{new}) should be higher then the lexical coverage before compound splitting (LC_{old}).

$$\frac{LM_{old}}{TM_{old}} < \frac{LM_{new}}{TM_{new}} \tag{10.7}$$

For every constituent migration scenario, the change in lexical coverage can be computed with for every scenario its own equation. Below, these equations are listed. For simplicity, for every compound splitting it is assumed that the compound is decomposed into two constituents α and β only. Every equation produces a lexical coverage that can be compared with the lexical coverage before compound splitting. In the equations, $f(C)$ is the frequency of the compound, $f(b_{1,2})$ the frequency of the first two words that are *not* in the lexicon (buffer), and $f(L_{last})$, $f(L_{last-1})$ the frequencies of the last two words *in* the lexicon:

(1) **When the compound is in the lexicon**

(1A) and, both the α constituent and the β constituent are in the lexicon, it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old} - f(C) + 2 \cdot f(C) + f(b_1)}{TM_{old} - f(C) + 2 \cdot f(C)} \tag{10.8}$$

as the frequency of the compound is removed from the lexicon, two times the frequency of the compound goes to the constituents in the lexicon and, as the compound is removed from the lexicon, a word that previously was out of the lexicon (first entry in the buffer: b_1) can now enter the lexicon. The total amount of words is changed also: for every compound splitting it applies that the compound with a frequency $f(C)$ is removed, instead, the frequency of the two constituents increase.

(1B) when only one constituent (α in this equation) is in the lexicon, it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old} - f(C) + 2 \cdot f(C) + f(\beta)}{TM_{old} - f(C) + 2 \cdot f(C)} \tag{10.9}$$

as the compound is removed, two times its frequency goes to the constituents, one of which is the β constituent that enters the lexicon.

(1C) when neither the α constituent nor the β constituent are in the lexicon, it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old} - f(C) + 2 \cdot f(C) + f(\alpha) + f(\beta) - f(L_{last})}{TM_{old} - f(C) + 2 \cdot f(C)} \qquad (10.10)$$

as this time both constituents enter the lexicon, the last word in the lexicon (L_{last}) is pushed out.

(2) **when the compound is not in the lexicon**

(2A) and, both the α constituent and β constituent are in the lexicon, it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old} + 2 \cdot f(C)}{TM_{old} - f(C) + 2 \cdot f(C)} \qquad (10.11)$$

as no word is removed from the lexicon but the compound constituents are in the lexicon so they both "receive" the frequency of the former compound.

(2B1) one constituent is not in the lexicon (α in this equation):

(2B1a) and this constituent reaches the lexicon after compound splitting (α):

$$f(\alpha) + f(C) > f(b_1) \qquad (10.12)$$

when the frequency of the compound is added to the frequency of the constituent and the resulting frequency is larger than the frequency of the first word not in the lexicon (b_1), it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old} + 2 \cdot f(C) + f(\alpha) - f(L_{last})}{TM_{old} - f(C) + 2 \cdot f(C)} \qquad (10.13)$$

as both constituents are in the lexicon, two times the compound frequency is added. The α constituent was not in the lexicon before so its frequency must be added as well. The last item in the lexicon is removed.

(2B1b) and the constituent does not reach the lexicon after compound splitting (α):

$$f(\alpha) + f(C) <= f(b_1) \tag{10.14}$$

when the sum of the frequencies of the compound and the constituent is not enough to reach the lexicon, it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old} + f(C)}{TM_{old} - f(C) + 2 \cdot f(C)} \tag{10.15}$$

as only one constituent in the lexicon "receives" the compound frequency.

(2C) neither the α constituent nor the β constituent is in the lexicon:

(2C1) and both α and β reach lexicon:

$$f(\alpha) + f(C) > f(b_1) \wedge f(\beta) + f(C) > f(b_1) \tag{10.16}$$

when the added frequencies of constituent and compound are large enough to let the constituents reach the lexicon, it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old} + 2 \cdot f(C) + f(\alpha) + f(\beta) - f(L_{last}) - f(L_{last-1})}{TM_{old} - f(C) + 2 \cdot f(C)} \tag{10.17}$$

as both α and β constituent reach the lexicon the compound frequency is added twice, along with the frequencies of the constituent but due to the entering of the constituents, the frequencies of the last two words in the lexicon must be removed.

(2C2) only one constituent (α) reaches the lexicon:

$$f(\alpha) + f(C) > f(b_1) \wedge f(\beta) + f(C) <= f(b_1) \tag{10.18}$$

when the added frequencies of constituent and compound are large enough for only one constituents (α) to reach the lexicon, it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old} + f(C) + f(\alpha) - f(L_{last})}{TM_{old} - f(C) + 2 \cdot f(C)} \tag{10.19}$$

as only one constituent reaches the lexicon, the compound and constituent frequency is only added once and one word has to be removed from the lexicon.

(2C3) neither one of the constituents reaches the lexicon:

$$f(\alpha) + f(C) <= f(b_1) \wedge f(\beta) + f(C) <= f(b_1) \tag{10.20}$$

when the added frequencies of constituent and compound are not large enough for either constituents to reach the lexicon, it applies that the old lexical coverage should be smaller than:

$$\frac{LM_{old}}{TM_{old} - f(C) + 2 \cdot f(C)} \tag{10.21}$$

as nothing changes in the lexicon.

Using these equations the effect on lexical coverage of compound splitting can be computed for every word. It must however be noted that the lexical coverages obtained using these equations is not entirely realistic. Locally, compound splitting may result in a lower lexical coverage whereas globally, compound splitting the particular compound may be beneficial. Such a scenario occurs for instance when compounds that share the same constituent(s) on their own do not improve coverage but taken together they do. However, as evaluating compound splitting globally is complex and computationally expensive (after every compound splitting the word frequency list must be re-ordered), it was decided to do a local evaluation. For this purpose, the word frequency file created out of the 1999-2001 newspaper data was used. First, the lexical coverage of a 65 K vocabulary before compound splitting was computed. Next, the word frequency list was descended and when a compound word was detected, the lexical coverage after compound splitting the compound was computed using one of the equations above and compared with the lexical coverage of the 65 K vocabulary. All compounds were labelled with a scenario type and either a “true” or “false” for a lexical coverage improvement and lexical coverage deterioration respectively. All compounds marked with a “false” label were then removed from the compound splitting conversion table to create a restricted compound splitting table. If the local evidence that a particular compound should not be decomposed for better lexical coverage results is valid, a global compound splitting run as conducted in the beginning of this section, this time with the restricted table instead, should yield better results.

Local lexical coverage statistics were computed for 290 K compounds in the word frequency list having only two constituents. Only 33 K received

scenario		number of "false" labels
1C		464
2C		2879
	2C1	6
	2C2	90
	2C3	2783

Table 10.4: Scenarios (1C, 2C1, 2C2, 2C3) with the number of compound splittings that did not yield a lexical coverage improvement according to the equations.

a "false" label, meaning that compound splitting had a negative local effect on lexical coverage. Most of the "false" labels were received due to scenario 2C: the compound is not in the lexicon and splitting pushes one former lexicon word out of the lexicon (see Table 10.4). The compounds with the "false" labels were removed from the compound splitting conversion table and the resulting table was used to decompound the 1999-2001 newspaper text data. The decomposed text data was compiled into a word frequency file that in turn was used to compute lexical coverages of vocabularies cumulatively, starting with taking only the first word in the word frequency file ($N = 1$) and ending with taking all words in the word frequency file ($N > 1\,M$). The results were compared with the unrestricted compound splitting version. In Figure 10.4 the differences in percentage lexical coverage between restricted and unrestricted compound splitting are plotted. The figure shows that lexical coverages are only slightly better for smaller vocabularies and become worse with larger vocabularies, although the difference is marginal. Given the small effect of this restricted method based on compound splitting scenarios and the computational effort that is needed for the creation of a compound exception list, it was decided to abandon this approach.

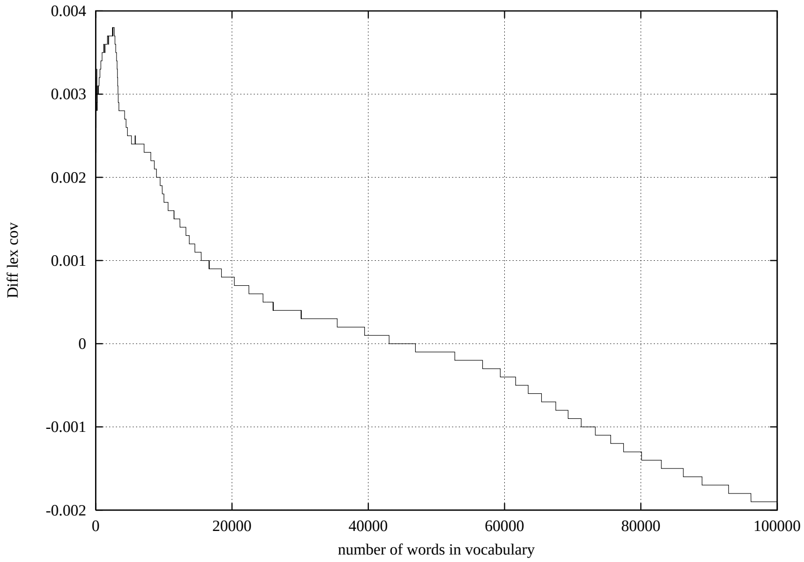

Figure 10.4: Differences in percentage lexical coverage between restricted and unrestricted compound splitting as a function of vocabulary size.

10.5 Speech recognition evaluation

10.5.1 Method

To investigate the effect of compound splitting on ASR performance, language models with a 65 *K* vocabulary (top 65 *K* most frequent words) were created based on different text versions of a Dutch newspaper collection of more than 300 M words. The original data set served as training data for the baseline language model, a number of differently decomposed text versions were used for the test language models. Compound splitting was done:

1. using an unrestricted compound splitting procedure:
   (a) treating the binding morpheme as a separate constituent.
   (b) attaching the binding morpheme to the preceding constituent (Glue-S).
2. Using restricted compound splitting procedures. Compounds were only decomposed if their frequency of occurrence was too low to be included in top *N* most frequent words in the original data where *N* was chosen to be 5 *K*, 20 *K* and 65 *K*.

The restricted procedure was created to investigate whether excluding frequent and probably well-modelled words could improve ASR performance over an unrestricted compound splitting procedure. Tri-gram Witten-Bell discounted backoff language models were created using a 65 *K* vocabulary of the most frequent words in the subsequent text data sets. Word pronunciations were obtained using a background pronunciation lexicon and a grapheme-to-phone conversion tool (see Chapter 4). As the grapheme-to-phoneme conversion may produce incorrect transcriptions, the pronunciation of words that were not included in all vocabularies (e.g., only occurred in one vocabulary), were manually checked to avoid that language model versions were put at a disadvantage as more words have to be produced by the grapheme-to-phoneme conversion tool, hence may have more incorrect word pronunciations.

The speech recognition system used acoustic models trained forward and backward in time on broadcast news training data (TNO-BN corpus, see Chapter 5. For the broadcast news transcription task, a collection of 18 Dutch broadcast news programs (*NOS Acht uur journaal*) recorded from January–March 2002 were transcribed manually. Segments containing non-speech or speech of a foreign language were excluded from the test data, resulting in approximately 6.5 hours of Dutch speech (70*K* words).

On comparing word error rates

The comparison of systems that apply compound splitting with systems that do not, deserves special attention. As noted in Carter et al. (1996), WER of systems applying compound splitting can be measured in two ways:

1. by taking the speech recognition hypothesis after compound splitting and comparing this with a reference in which compounds are split as well (split comparison),

2. by mapping the compound constituents in the speech recognition hypothesis after compound splitting back to the original compounds, and comparing this with the original reference (unsplit comparison).

One could argue that for a fair system comparison the total number of words to be recognised correctly should be the same. In that case, the WER computations for the respective systems should follow the same procedure: when method [1] was chosen for WER computation of the system that applied compound splitting (split system), also a split comparison had to be performed for the system that did not apply compound splitting (unsplit system). This means in practice that compounds in the hypothesis transcription of the unsplit system should be split afterward for a comparison with the split reference used in method [1]. Alternatively, method [2] is applied for WER computation of both systems. However, as pointed out by Carter et al., mapping compound constituents in the split system's hypothesis back to the original compounds, may introduce errors as this method creates a compound for every sequence of words that in some context could be compound. For example, the word sequence "*modder smijten* (English: throwing mud)", could in Dutch be a compound in "*het moddersmijten is begonnen* (English: the mud-slinging has started)" or two words in "*hij begon met modder smijten* (English: he started throwing mud)".

Another problem with applying method [2] stems from the shortcomings of the compound splitting procedure that was used. This table was created on the basis of newspaper data that in some cases contains word sequences that miss the required space, such as in "*peilingenvoorspellen* (English: the polls predict)", that should have been written as "*peilingen voorspellen*". When such flaws appeared frequently enough in the data, they could well have been included in the compound splitting table, as "*peilingen*" en "*voorspellen*" are valid Dutch words. Up until this point, this was not recognised as being problematic as splitting such incorrect compounds could be regarded as a welcome extra text correction step. However, when applying method [2] the occurrence of these entries in the compound splitting table introduces errors.

Given the errors that could be introduced when applying method [2], for this experiment it was chosen to compare the unsplit system with the split systems as follows:

- by comparing WERs that were computed using method [1] for both the split and unsplit systems.
- by comparing the WER of the split systems computed using method [1] with the WER of the unsplit system computed using the original unsplit reference.

The last comparison method (further referred to as method [3]) does not actually provide a fair comparison in terms of speech recognition performance. It was however included as it can be argued that with this method split systems and unsplit systems can better be compared from a language processing point of view. This method was also regarded to be of primary relevance in the study of Carter et al. (1996).

10.5.2 Results

	OOV	WER (split)	WER (unsplit)
unsplit system:			
baseline 65 *K*	2.59	39.8 %	39.5 %
split systems:			
unrestricted 65 *K*	2.18	39.2 %	
unrestricted+glueS 65 *K*	2.22	39.2 %	
restricted 65 *K*	2.25	39.6 %	
restricted 20 *K*	2.19	39.1 %	
restricted 5 *K*	2.18	39.1 %	

Table 10.5: OOV rates and WER rates of an unsplit system and split systems: unrestricted refers to a compound splitting procedure that splits all encountered compounds, restricted refers to procedures that split compounds only when they do not occur in the top *N* words of a sorted word frequency list. The “unrestricted-glueS” refers to the method that attaches the binding morpheme “s” to the preceding constituent.

Table 10.5 lists the OOV rates and WER rates of the baseline, unsplit, system and the various split systems using language models based on decomposed text versions. It shows that the split systems all perform better than the baseline. OOV rates drop with a maximum of almost 16 % relative (unrestricted 65 *K* and restricted 5 *K*). Using the split comparison (method [1]), the highest performance gain of 1.8 % relative was obtained using a restricted compound splitting procedure (splitting only compounds that do not appear in the top 20 *K* or 5 *K* in the sorted word frequency list). Although the OOV rate slightly increased, no effect on WER was observed for the compound splitting procedure that attaches the binding morpheme “s” to the preceding constituents.

10.5.3 Conclusion

The results demonstrate that using decomposed text data for language model training improves the coverage of speech recognition vocabularies and as a result of that, speech recognition performance, regardless of possible disturbing side-effects of compound splitting as mentioned in the introductory section. No effects could be observed caused by a different treatment of the binding morpheme “s”. The hypothesis that one should not alter compounds that are highly frequent as they will probably have robust n-gram probability estimates, was confirmed by the experiment. The results suggest that omitting compounds in the 0-20 *K* word frequency range is sufficient for optimal performance. Noteworthy is the fact that the WER of the 20 *K* restricted model equals the one of the 5 *K* restricted model, although its OOV rate is slightly worse. This may indicate that the negative effect of having more OOV words is neutralised by more robust n-gram models. As the performance difference between the best performing language model and the language models based on unrestricted compound splitting was only marginal (1 % absolute), the computationally less expensive procedure of unrestricted compound splitting may be preferred in practice.

10.6 Summary and final conclusions

To investigate the effect of compound splitting on speech recognition performance, a data-driven compound splitting algorithm was created that used an alphabetically sorted word frequency file based on a large amount of newspaper data. Although recall of the algorithm could not be computed, it was explained that the design of the algorithm guarantees that compound recall will at least be sufficiently high for the purpose of the experiment. Precision of the algorithm, measured on the first iteration of the algorithm and using a commercial dictionary with constituent boundary labels, was 97,7 %.

A first comparison between the original text version and a fully decomposed text version showed a 20 % reduction in distinct words and a better lexical coverage for lexicons derived from the decomposed data. Restricting the compound splitting procedure aiming at even better lexical coverage performance was investigated in detail by looking at the contribution to the self-coverage of a 65*K* lexicon of individual decompositions. When such individual decompositions did not improve lexical coverage by themselves, they were excluded from the list of compounds used for the actual decomposition of the text data. However, this procedure did not yield a robust improvement of lexical coverage of lexicons.

In order to investigate the effect of compound splitting on speech recognition performance, a number of language models were created. One was based on the original text data (baseline) and the others were based on text data that was decomposed in different ways: two using a full compound splitting procedure without applying any restrictions, and three using a restricted compound splitting procedure that only decomposed compounds when they were not frequent enough to be included in a lexicon of size N, were N was chosen to be $5\,K$, $20\,K$ and $65\,K$. For the unrestricted procedure, two text versions were created: one that treated the binding morpheme as a separate constituent and one that attached the binding morpheme to the preceding constituent. The language models were evaluated in a broadcast news transcription task. The language models that were created using the restricted compound splitting procedure, in which compounds in the 0-$20\,K$ word frequency range were omitted, gave the best speech recognition performance. However, the performance difference with the language models based on unrestricted compound splitting was only marginal (1 % absolute).

Although the research described in this chapter confirmed the hypothesis that compound splitting can improve lexical coverage and speech recognition performance for Dutch, some issues remain that deserve to be looked into more closely. Firstly, for specific tasks, having available an accurate reverse compound splitting procedure (that maps constituents back to the original compounds as addressed in Section 10.5.1) may be crucial. In a spoken document retrieval framework, reverse compound splitting is of minor importance (see also the discussion in the introductory section), but in a dictation task for example, it may be expected that a user of a speech recognition system requires compounds to be reassembled automatically. Applying reverse compound splitting in this research introduced a substantial amount of incorrect compounds. In order to improve this procedure, the compound detection algorithm could be improved so that incorrect compound mappings can be prevented. Alternatively, the compound splitting table can be post-processed to exclude entries that introduce incorrect mappings.

A second issue for future research is the exact behaviour of compounds and decomposed compounds in the language model. Given that WER improved with increasing OOV rates using language models based on the restricted splitting procedures, it was hypothesised that this procedure enables the creation of more robust n-gram language models. However, this research could not provide evidence for this hypothesis. It may therefore be worthwhile to search for an experimental design that enables the investigation of compound splitting on the language model level. Furthermore, the effects of applying the opposite of compound splitting could be included in this investigation: the combination of frequent orthographic word tuples, referred to as multi-words, into single items in the recognition lexicon, as proposed for example by Gauvain et al. (1997).

In summary, it can be concluded that:

- when large text collections are available, a data-driven compound splitting procedure is a simple but effective approach for the generation of a compound splitting table.
- compound splitting reduces the number of distinct words in a Dutch text collection and enables the generation of vocabularies with a better coverage compared to vocabularies generated from the original data.
- the hypothesis that a restricted compound splitting procedure results in additional lexical coverage improvements, could not be warranted.
- language models based on decomposed text data result in a small but consistent Dutch speech recognition performance improvement in comparison with standard language models.
- the best speech recognition performance was obtained when language models were used that were based upon a restricted compound splitting procedure that only splits compounds that are relatively infrequent: the ones that do not occur in the 0-20 K word frequency range.
- to enable a correct reverse compound splitting procedure, either adapting the current compound detection algorithm or post-processing the compound splitting table is required.
- to obtain a better understanding of the effect of compound splitting on the robustness of n-gram language models, additional research is required.

Chapter 11

Speech recognition evaluation

The main focus in this chapter is on the language modelling part of the speech recognition system. A number of language model creation procedures are evaluated with regard to speech recognition performance in a full scale broadcast news transcription task. As a number of issues from previous chapters are involved in these evaluations, this chapter may also be viewed as the final evaluation of the task of porting the ABBOT system to Dutch.

11.1 Introduction

In the previous chapters, a number of issues regarding the porting of the English *ABBOT* system to Dutch were addressed, including the collection of Dutch training data for acoustic modelling and language modelling, the generation of Dutch word pronunciations, the training of Dutch acoustic models, text normalisation for language modelling, language model vocabulary selection and compound splitting. In this chapter, most of these issues are brought together in a set of speech recognition evaluations that primarily focus on the language modelling part of the speech recognition system, but at the same time give an impression of their application and effect in a full scale broadcast news transcription task as well. The evaluations reported in this chapter may therefore also be viewed as the final evaluation of the task of porting the *ABBOT* system to Dutch, that as such provides an indication of the final performance of the system in a broadcast news transcription task as a result of the research described in this thesis. The vocabulary selection strategy, proposed in Chapter 9, and compound splitting (previous chapter) were however not evaluated again with the experiments described here.

Language modelling for the broadcast news transcription tasks is well-studied, especially for the English language. From the NIST/DARPA Broadcast News (Hub4) speech recognition benchmark tests (Pallett, 2002), ample experience with language modelling in this domain was obtained. This experience was reported in numerous papers—such as the special issue on broadcast news speech recognition of *Speech Communications*[1]—that have served as a guideline for the choice of language modelling techniques to be investigated in this research. Given the available data, language modelling tools, speech recognition architecture, and time, it was however not feasible to study all promising or successful techniques. For example, although increasing the length of the n-gram context to 4-grams or even 5-grams is reported to reduce perplexity and word error rates (e.g., Sankar et al., 2002), this could not be evaluated for Dutch as the available *ABBOT* decoders, *CHRONOS* (Robinson and Christie, 1998) and *NOWAY* (Renals and Hochberg, 1999), are limited to the use of 3-gram language models[2]. The creation of language models was merely studied pragmatically, aiming at obtaining the best possible speech recognition performance for the task domain, given the available resources and time, so that the speech recognition system could successfully be deployed in a spoken document retrieval task. In addition and if appropriate, it was attempted to relate the results of the experiments described in this chapter to the results reported in the literature on language modelling in the context of broadcast news transcription.

In Chapter 6, the basics of n-gram language modelling were already discussed as a reference for the following chapters. As for the issues that are not specifically addressed in the evaluation section below—such as smoothing techniques—the reader is referred to this chapter. In the next section, first the general experimental set-up is described. Next, language model procedures are described and evaluated chronologically: starting from scratch by investigating a few general aspects of the procedure (baseline experiments) and ending with the investigation of more sophisticated techniques (data selection and mixture language models). In Section 11.5 the results are summarised and discussed both from a language modelling and general speech recognition point of view.

11.2 Experimental setup

Text data

The normalised newspaper data from January 1999 until December 2001 served as main training corpus for the language models in the experiments. Unless reported otherwise, no topic selection was applied, and all available newspaper data was used for the generation of n-gram counts. For a number of experiments, the data set was augmented with the autocues data from the same time period. Given that the test data was from 2002, it was

[1] Speech Communications, Volume 37, 2002

[2] The srilm-enhanced version of the NOWAY decoder, developed at ICSI by Chuck Wooters, can also use 4-gram models

chosen not to use recent data (2002 onward) to simulate an *speech recognition!online* recognition task, for which evidently no future data is available, as opposed to a *retrospective* recognition task, that is performed after all data is recorded and allows for the use of future data to optimise the language models (Auzanne et al., 2000). The text data collection was described in detail in Chapter 7, the normalization procedure in Chapter 8.

Test data

A set of 10 broadcast news shows from January–March 2002 were transcribed manually on the word level and also manually segmented in sections and sentences. The sections correspond with the individual topics in the news shows and every section consist of a varying number of sentences. The topic segmentation was only significant in a few experiments. Segments containing non-speech or speech of a foreign language were excluded from the test data. In total, the test data contained approximately 3 hours of Dutch speech (35 K words, on average 257 words per section).

Word pronunciations

The pronunciations of the vocabulary words were obtained by consulting the GVD background pronunciation lexicon of 1.3 M words. If the background lexicon could not provide a transcription, the pronunciation was generated automatically by the GVD grapheme-to-phone converter (G2P). As for 65 K vocabularies on average 25 % of the word pronunciations had to be produced by the G2P tool, it was not feasible to check these manually. In Chapter 4 the specifications of the background lexicon and G2P tool can be found.

Acoustic modelling

The speech recognition set-up used acoustic models trained forward and backward in time on broadcast news training data of 2000 (TNO-BN, 256 state units) as described in Chapter 5.

Assessment

A number of scoring statistics were evaluated in the language model experiments:

1. *word error rate* (WER) and *mean story word error rate* (MSWER)

   Speech recognition performance was measured by computing the overall word error rate (WER) and the mean of the word error rate per section (MSWER), which is referred to as “mean story word error rate”. The sections in the broadcast news material can each be viewed as a separate “story” or document. In spoken document retrieval evaluations it is customary to provide WERs per story, in order to relate retrieval performances per query to recognition performance of the retrieved documents. Moreover, the MSWER is a more realistic measure for unbalanced test material as (very) low performances on specific acoustic conditions are given less weight.

2. *perplexity* (PP)

   In order to relate speech recognition performance to language model performance, the perplexities of the respective LMs were computed using the reference transcripts of the 10 broadcast news shows in the test collection containing $35K$ words. During LM development one does usually not measure perplexity on the test data itself in order to prevent that the speech recognition results are biased. In this evaluation the given perplexities only provide a measure to relate speech recognition performance to LM performance. During the actual LM development, perplexities were obtained using another set of broadcast news transcriptions from the year 2000 (BN2000test).

3. *out-of-vocabulary rate* (OOV)

   As was extensively discussed in earlier chapters, the out-of-vocabulary rate (OOV) of the speech recognition dictionary is to a large extent determinant for speech recognition performance. Therefore, the actual OOV rate of the recognition dictionary on the test data is reported where appropriate.

4. *G2P contribution* (G2P)

   In Chapter 4 it was noted that word pronunciation errors may be introduced when a G2P tool is used for the generation of word pronunciations. When the speech recognition dictionary is changed in the evaluations, therefore also the contribution of the G2P to the word pronunciation generation process is given.

5. *language model size* (LMsize)

   The size of a language model can be significant, especially for online recognition tasks, as it influences memory usage and computation time. In specific cases, a smaller size LM that performs slightly worse, may be preferred over a larger LM. To provide an indication of language model size differences given specific LM configurations, the size of the (compressed ARPA format[3]) LM is provided if appropriate.

[3] The ARPA-standard language model format was introduced by Doug Paul and is commonly used in speech recognition research

Software

As creating language models using large amounts of data and state-of-the-art LM algorithms puts high demands on a software implementation, existing language modelling software was used for this research. Two software packages for language model training can be freely obtained and are widely used in the speech recognition research community: the *Cambridge-CMU language modelling toolkit* (CCLM) (Clarkson and Rosenfeld, 1997) and the *SRI language modeling toolkit* (SRILM) (Stolcke, 2002) (see Appendix C.2 for a short description of the toolkits). Although each package has its advantages and shortcomings—CCLM for example is more efficient in memory usage, whereas SRILM provides better interpolation options—all language models in this experiment were created using the SRI language model toolkit for compatibility reasons. The choice for a single toolkit implies that the language model procedures that are investigated here are to some extent limited to the options that are provided by the toolkit, which especially applies for the discounting schemes. SRILM does not support the Jelinek-Mercer type of deleted interpolation or maximum entropy modelling, so only Katz back-off smoothing was applied.

With the *NOWAY* decoder the ARPA-format produced by the language model toolkit was converted to the binary format required by the *CHRONOS* decoder, that was actually used for word decoding (see also Section 3.3.4). Scoring of the speech recognition hypotheses was done using the *sclite* scoring software described in Appendix C.2.

Information retrieval techniques

For some of the language model creation procedures, information retrieval (IR) techniques were deployed using the Okapi *tfidf* term weighting scheme according to Robertson et al. (1998):

$$\sum_{T\in Q} w \frac{(k_1+1)}{K+tf}\frac{(k_3+1)qtf}{k_3+qtf} + k_2 \cdot |Q| \cdot \frac{avdl-dl}{avdl+dl} \tag{11.1}$$

where Q is a query containing terms T, w is the Robertons/Spärck Jones weight (Robertson and Spärck-Jones, 1976) of T in Q:

$$w = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \tag{11.2}$$

N, number of items in the document collection
n, number of documents containing the term
R, number of documents known to be relevant to a specific topic
r, number of relevant documents containing the term
S, number of documents known to be non-relevant to a topic
s, number of non-relevant documents containing the term
K is $k_1((1-b)+b\cdot dl/avdl)$.

The values chosen for the parameters k_1,b, k_2 and k_3 depend on the nature of the queries and on the data collection. For the research described in this thesis, the following parameter settings were chosen:

$$\begin{aligned} k_1 &= 1.2 \\ b &= 0.75 \\ k_2 &= 0 \\ k_3 &= 1000 \end{aligned}$$

With $k_2 = 0$, equation 11.1 can be written as:

$$\sum_{T \in Q} w \frac{(k_1 + 1)tf}{K + tf} \frac{(k_3 + 1)qtf}{k_3 + qtf} \tag{11.3}$$

11.3 Baseline experiments

In this section, the results of a number of experiments are reported that investigate general aspects of the language modelling procedure, including vocabulary size, n-gram cutoffs and smoothing techniques. The goal of conducting these experiments was to arrive at the most optimal baseline settings for language modelling in the Dutch broadcast news domain, given the available resources.

11.3.1 Vocabulary size

In Chapter 9 the speech recognition vocabulary was studied from an out-of-vocabulary rate perspective. Vocabulary size was regarded as a constant that is limited to $65K$ words given the available decoders. The focus in that chapter was on making the best possible selection of words given this vocabulary size limit. However, it was also briefly mentioned that acoustic confusability becomes more probable with larger vocabularies as the number of words that differ only in a few phones grows. Rosenfeld (1995) estimated that the optimal vocabulary size for large vocabulary tasks to be roughly between $55K$ and $110K$ words.

In order to investigate the effect of growing vocabularies on speech recognition performance in a Dutch broadcast news transcription task as a function of vocabulary size, five language models were created that only differed in the size of the vocabularies that were used: $5K$, $10K$, $20K$, $40K$ and $65K$ words. A standard Katz back-off language modelling procedure was performed, using Good-Turing (gt) discounting with lower exclusion cutoffs for the bigrams and trigrams with counts of 2 (lower counts are discounted to 0) and lower and upper discounting cutoffs for all n-grams with counts larger then 7.

Vocab	LM size	PP	OOV	WER	MSWER	G2P
5 K	54	87	12.89 %	50.2 %	50.9 %	0 %
10 K	73	114	8.02 %	44.5 %	43.9 %	1.52 %
20 K	91	135	5.13 %	40.7 %	39.2 %	8.04 %
40 K	107	155	3.00 %	38.0 %	36.2 %	15.98 %
65 K	116	149	2.05 %	37.1 %	35.0 %	23.23 %

Table 11.1: Language model size, perplexity, OOV rate, word error rates and G2P contribution as a function of language model vocabulary size.

Results

Table 11.1 shows the statistics for the vocabulary size runs. As expected, the larger vocabularies reduce the number of OOV words from almost 13 % for the 5 K vocabulary to some 2 % for the 65 K vocabulary. The decreasing OOV rate results in an improved speech recognition performance, in spite of the increased acoustic confusability and the growing amount of less reliable pronunciations produced by the G2P. The results, visually depicted in Figure 11.1, show an almost a linear relationship between OOV rate and overall word error rate with almost an equal word error reduction with every percent OOV reduction.

11.3.2 Cutoffs

N-grams that occur only once in the training data (singletons) are regarded as unreliable and are usually excluded from the language model by setting a cutoff variable. Especially when the amount of training data is large, one could argue that n-grams that occur twice are almost as unreliable as those that occur once and that therefore only n-grams with a minimum count of 3 should be included in the language model. The cutoff parameter can also be used to reduce the size of the language model by excluding less reliable n-grams.

In this experiment a number of bigram and trigram cutoff-settings are tested on 65 K Good-Turing discounted Katz back-off language models: excluding singletons, and excluding n-gram with counts of respectively 2 (the 65 K run from the previous experiment), 3 and 5.

Results

In Table 11.2 the results of the respective language models are listed. Note that the OOV rate and G2P contribution were equal to the statistics for the 65 K vocabulary given in Table 11.1. The results show that although removing singletons only (cutoffs of 1) gives the best performance, using

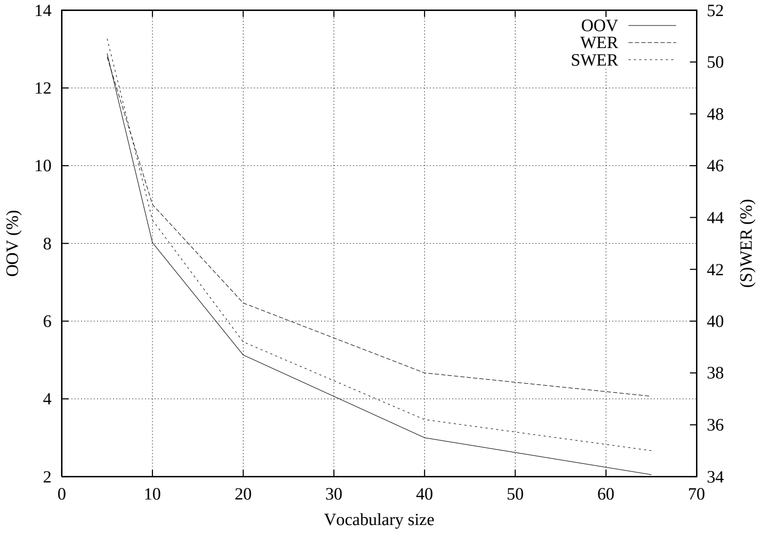

Figure 11.1: OOV rate, WER and MSWER as a function of vocabulary size

an n-gram cutoff of 2 is regarded as the best choice as this gives a LM that is 75 % smaller, has a better perplexity measure and hardly shows any performance degradation.

2/3-gram cutoffs	LM size	PP	WER	MSWER
1	414	161	37.1 %	34.9 %
2	116	149	37.1 %	35.0 %
3	86	172	37.5 %	35.5 %
5	39	183	37.8 %	35.9 %

Table 11.2: Language model size, perplexity and word error rate as a function of n-gram cutoffs.

11.3.3 Smoothing methods

Many techniques have been proposed for n-gram smoothing and some widely used techniques were described in Section 6.3. In Chen and Goodman (1998) a number of smoothing techniques are compared as a function of data size, corpus, cutoffs and n-gram order, as these variables have shown to be significant for the performance of the smoothing techniques.

Also, the effect on speech recognition performance of the smoothing algorithms was investigated: better smoothing algorithms were reported to yield up to a 1 % absolute difference in word error rate. Although it does not seem a major improvement, one must consider that in order to improve general speech recognition performance, every small improvement that could be obtained anywhere in the process should be gathered.

Chen and Goodman found that (modified) Kneser-Ney smoothing consistently outperforms other smoothing algorithms over all training set sizes and corpora. Kneser-Ney smoothing (kn) was therefore an obvious first choice for this experiment. In the same study, it was reported that absolute discounting and Witten-bell discounting methods have a very low performance on small data sets, but yield better results for very large data sets. As the available Dutch training data of 300 M words in the terminology of the Chen and Goodman study may be regarded as very large (comparable to the 10 M sentences with on average 21 words per sentence of the WSJ/NAB corpus in Chen and Goodman (1998)), also Witten-bell discounting (wb) and absolute discounting (abs) were chosen for this experiment. As a reference discounting method, the default discounting scheme provided by the SRI language modelling toolkit and used in the previous experiments, Good-Turing discounting (gt), was chosen.

SRILM provides a special feature to create interpolated backoff models that are reported to yield slightly better results than the standard models (Stolcke, 2002). As this feature could easily be implemented, for every language model also a interpolated version (ipl) was created except for the one that applies Good-Turing discounting as interpolation is not supported here. All language models used Katz backoff for smoothing, n-gram cutoffs of 2 and the same 65 K vocabulary as used in the previous runs.

Results

Table 11.3 shows the perplexities and word error rates of the different discounting methods. Although Good-Turing discounting has the best overall perplexity, Kneser-Ney discounting with interpolation gives the best speech recognition performance with the lowest MSWER of 34.6 %. However, the differences between the discounting schemes are marginal: the highest and lowest MSWER differ only 0.4 % absolute which is less than the 1 % absolute difference that was hoped for. In spite of that, Kneser-Ney discounting with interpolation was regarded as the best discounting method in the current experimental setup and used as such in the following experiments.

11.3.4 N-gram pruning

In Sankar et al. (2002) and Stolcke (1998), a method for reducing the language model size is reported that prunes n-grams based on the minimal distance (relative entropy) between the probability distribution of the original model and the distribution of a pruned model: all n-grams that change

Discounting method	PP	WER	MSWER
kn+ipl	159	36.9 %	34.6 %
abs	161	36.9 %	34.9 %
abs+ipl	161	36.9 %	34.9 %
wb+ipl	160	37.0 %	34.8 %
wb	164	37.1 %	34.9 %
gt	149	37.1 %	35.0 %
kn	168	37.3 %	34.9 %

Table 11.3: Language model perplexity and word error rates as a function of smoothing techniques with (ipl) and without interpolation approximation.

the perplexity of the pruned model by less than a threshold are removed. Although a language model size of around the 100 Mb, as in these experiments, is not problematic, augmenting the training data set may lead to LM sizes that are impractical. Therefore, the effect of this pruning method was investigated on the Kneser-Ney discounted language model of the previous experiment, using a pruning thresholds of 10^{-7}, 10^{-8} and 10^{-9}, the same threshold as were tested in Sankar et al. (2002).

Results

The results of the pruning evaluations are shown in Table 11.4 that lists the language model size, the number of bigrams and trigrams, perplexity and word error rates as a function of the pruning threshold. It shows that delicate pruning, using a threshold of 10^{-9}, reduces the number of n-grams and the size of the language model substantially, with only minor performance degradation (0.1 % absolute). Comparing these results with those reported in Sankar et al. (2002) however, shows that pruning in this experiment is less effective: even a moderate pruning with a threshold of 10^{-8} did not lower recognition performance in the study of Sankar et al., whereas in this experiment the performance degradation is substantial. It must be noted however that in Sankar et al. (2002) 4-grams instead of 3-grams were evaluated.

11.3.5 Conclusions

The goal of the experiments described hitherto was to investigate the optimal baseline settings for LM creation in the context of Dutch broadcast news transcription tasks, given a variety of LM properties proposed in the literature. It was shown that in line with the results obtained for languages as English, increasing vocabulary size up to a maximum of $65\,K$ words is

Thresh	LMs	#bigrams	#trigrams	PP	WER	MSWER
no prun	112	6,226,759	16,576,907	159	36.9 %	34.6 %
10^{-9}	94	5,715,941	12,866,842	161	37.0 %	34.7 %
10^{-8}	56	4,410,050	5,681,035	170	37.5 %	35.2 %
10^{-7}	16	1,395,331	999,948	210	39.8 %	38.0 %

Table 11.4: Language model size, number of bigrams and trigrams, perplexity and word error rate as a function of n-gram pruning.

beneficial and improves Dutch speech recognition performance, in spite of the fact that acoustic confusability increases as well. The outcome of the vocabulary experiment suggest that the most appropriate vocabulary size for a Dutch broadcast news transcription task, is at least larger than the $40\,K$–$60\,K$ range estimated by Seymore et al. (1997) in a comparable task for English. Most likely this is the result of the larger lexical variability of Dutch compared to English. As the maximum vocabulary size of the *AB-BOT* system was limited to $65\,K$ words, speech recognition performance as a function of even larger vocabularies, could not be tested.

The experiments that addressed the optimal cutoff-settings showed comparable results as can be found in the language modelling literature: the larger the cutoff-settings, the smaller the language models become, at the cost of speech recognition performance. Neglecting all n-grams with a frequency of 2 was in this research regarded as the most optimal choice. Although speech recognition performance dropped slightly, the LM was 75 % smaller and had a better perplexity than the model that only removed singletons (cutoff of 1). Applying n-gram pruning in order to obtain even smaller LMs without sacrificing too much in performance, was reasonable successful. Using a delicate pruning scheme reduced the size of the LM substantially at the cost of only a minor loss in performance. In line with the findings for English of Chen and Goodman (1998), Kneser-Ney smoothing with interpolation reached at the highest speech recognition performance in the experiment, although the difference with other smoothing methods was only marginal.

11.4 Data selection and language model adaptation

Up until this point, all available newspaper data was used for the creation of language models. However, some newspaper articles have only little in common with the content in the broadcast news domain. Articles about chess, cooking recipes and the television guide are evident examples of off-domain text data that possibly introduce an uncertain amount of noise in the language models. Although the n-grams encountered in such articles may be valid Dutch n-grams, they may well be improbable in a broadcast news context (as for example in “he lost his queen”). Therefore, it was investigated whether the topic and category labels of the newspaper corpus (e.g., “sports”, “economics”, see also Section 7.2) could be deployed to exclude the less representative text data from the language model creation procedure. Therefore, in a supervised classification procedure, a finite set of broad text clusters such as “foreign affairs”, “politics”, “disasters” and “health” was created on the basis of a much broader set of topic labels.

In a first experiment, sets of clusters were selected that intuitively seemed the most appropriate clusters for the broadcast news domain for language model training. However, none of the cluster selections could improve the perplexity that was obtained using the full data set: the best performing selection resulted in a perplexity of 173, whereas the full data set reached at a perplexity of 159. In a second experiment, an initial language model was created by starting with a single cluster as training data. In the following steps, a new cluster was added to the training data and a new language model was created. If the newly created language model gave a lower perplexity on a set of broadcast news transcripts of the year 2000 (BN2000test) than the previously created LM, the added cluster was marked as “useful”. Eventually, all useful clusters were taken together for the generation of a single language model. However, the language model that was trained on the full data set still gave the lowest perplexity. Apparently, given the available amounts of training data, data quantity (the more data the better) is still more important than data quality (selecting the most appropriate data) in language model estimation. Another explanation might be that the quality of the selected data was not yet high enough to allow for language model improvements.

Deploying mixtures of language models, for example a mixture of a domain specific language model based on a small amount of domain specific data and a general model based on the full data set, is often proposed to avoid problems of data sparsity in the context of data selection for language modeling (e.g. Clarkson and Robinson, 1997; Gotoh and Renals, 2000; Seymore and Rosenfeld, 1997). The next sections report the results of the creation of a number of mixture language models. First, the autocues data are exploited to create better language models for the broadcast news domain. The autocues data can be a valuable source for language modeling in the broadcast news domain as this data perfectly matches the newsreader’s parts in the broadcast news shows (see Chapter 7). Next, experiments that investigate the use of information retrieval techniques for the selection of domain specific data in a mixture language model approach are described.

11.4.1 Incorporating autocues data

Two approaches were investigated to incorporate the autocues data into the language modeling procedure. In the first approach, the autocues data was simply added to the total newspaper training collection. Next, a language model (backoff interpolated Kneser-Ney LM) was created on the basis of the complete data set. In a second approach, two separate language models were created. One on the basis of the newspaper data and the other on the basis of the autocues data. Next, both language models were linearly interpolated (Clarkson and Robinson, 1997) to generate a mixture language model. The mixture-weight (λ) was estimated using the text transcripts of the five broadcast news shows of the BN2000test set.

Results

Configuration	PP	WER	MSWER
kn-ipl	159	36.9%	34.6%
news+acues	154	36.7%	33.8%
mixlm-news+acues	147	36.4%	33.9%

Table 11.5: Perplexity and word error rates of a baseline LM based on newspaper data only (kn-ipl), a LM based on newspaper data plus autocues data (news+acues), and a mixture language model, created by interpolating a newspaper LM with a autocues LM.

The results in Table 11.5 show that adding domain specific data improves both perplexity and speech recognition performance compared to the baseline (kn-ipl). However, the best perplexity and best overall speech recognition performance is obtained when the domain specific language model is interpolated with a specific language model (mixlm-news+acues).

11.4.2 Topic based language models

An alternative approach to the data selection methods based on topic or category labels described earlier, is the use of information retrieval techniques to find the most appropriate selection of training data for a certain domain. Many of such approaches have been investigated, including topic detection using topic clustering and a topic classifier based on a model of word co-occurrence (e.g., Sekine and Grisham, 1995; Seymore and Rosenfeld, 1997), and latent semantic analysis (LSA, Bellegarda, 2000).

In this research, a straightforward but computationally expensive procedure was implemented as a first approach to the creation of topic based language models using IR techniques. Given the somewhat disappointing results of the clustering approach described above, the newspaper data collection was regarded as an unstructured set of documents, each having a particular topic that may or may not resemble a topic in a specific section in a broadcast news show. To find those documents in the newspaper collection that resemble a topic in a particular broadcast news section, first a preliminary word transcription of the broadcast news section was created using a baseline speech recognition configuration (it was assumed that section boundaries in broadcast news shows were known). Next, this word transcription was used as a query in an IR system in order to generate a ranked list of newspaper documents that match the query. This list of documents was then used to generate a topic based vocabulary and language model. In a second recognition run the final word transcription was generated. This procedure is visually depicted in Figure 11.2.

The IR system uses Okapi term weighting as described in Section 2.2 and a stop list. A number of experiments were performed to find the optimal settings. To investigate the effect of speech recognition errors of the initial ASR run (query) on retrieval and final speech recognition performance, three query conditions were distinguished:

- one based upon a perfect speech recognition output by taking the reference transcript as input for the query,
- one based upon a relatively good speech recognition performance by taking the output of a speech recognizer using a 65 K vocabulary and a Kneser-Ney backoff LM from earlier experiments (WER of 36.9 %, see the experiments on smoothing techniques above),
- and one based upon a relatively bad speech recognition performance by taking the output of a speech recognizer using a 5 K vocabulary and Good-Turing discounting (WER of 50,2 %, see the experiments on vocabulary size above).

Next to the query conditions, three conditions were created on the basis of the number of relevant documents that were used for topic based LM creation: 3 K, 5 K and 10 K documents. As on average a document contained 350 words, the language models were based on some 1 M, 1.75 M and 3.5 M words respectively. Given the discussion on data sparsity above, best results should be obtained using as many relevant documents as possible.

All vocabularies were created on the basis of the relevant documents that were used for LM creation by selecting all words with a minimum count of two from these documents. As the total number of distinct words in these documents was relatively small, none of these vocabularies reached

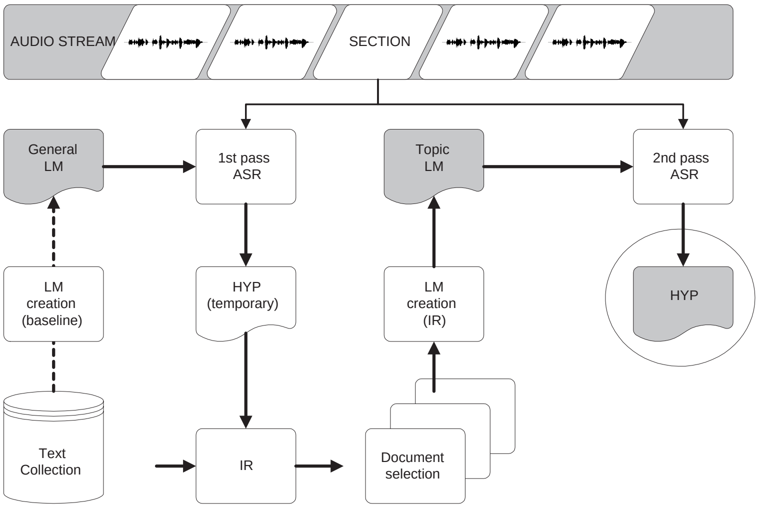

Figure 11.2: Creation of topic based LM: the word transcript of an initial decoding step using a recognition system with a baseline LM are used to generate a query. Given the query, the IR system produces a set of relevant documents that are used for creating the topic specific LM. In a second recognition run the final word transcript is generated.

the maximum of 65 K number of words. To investigate whether using all available vocabulary space could improve performance, a special condition was created: words from the 65 K vocabulary used in previous experiments that were not yet included in the topic LM vocabularies were added up to a vocabulary size of 65 K words, the most frequent words first.

Finally, a *mixture* topic LM condition was created that used the top 3 K most relevant documents for topic based LM creation. This topic LM was then interpolated with a general model based on newspaper data and autocues data (comparable with the mixlm-news+acues model of the previous experiment). This general model was based on a 40 K vocabulary in order to prevent that the vocabulary of the mixture LM could exceed the maximum vocabulary space of 65 K.

Results

LM	WER	MSWER	Dct	G2P
mixlm-news+acues	36.4%	33.9%	65,000	
irlm-top3K-bad	37.8%	35.0%	26,566	15.61%
irlm-top3K-base-65K	36.9%	33.3%	65,000	25.46%
irlm-top3K-base	36.9%	33.2%	25,464	15.33%
irlm-top5K-base	36.5%	33.0%	36,305	18.91%
irlm-top10K-base	36.2%	32.7%	56,360	24.21%
irlm-top3K-ref	36.1%	31.5%	25,112	15.10%
irlm-top3K-news+acues	35.3%	32.5%	40,730	16.73%

Table 11.6: Word error rates, average dictionary size and average G2P contribution as a function the different configurations applied for the creation of LMs using IR.

The results are listed in Table 11.6. As a reference the results of the mixture LM based on newspaper and autocues data are provided (mixlm-news+acues). Using a baseline speech recognition configuration for the first decoding pass and selecting the top 3 K documents for LM generation (irlm-top3K-base) already gave a speech recognition performance in the second pass that resembled the performance of the mixture LM, although the average dictionary size was substantially smaller (25 K instead of 65 K). Using a perfect speech recognition transcripts for querying (irlm-top3K-ref) even outperformed the mixture LM, whereas using an errorful transcript, represented by a LM with a small vocabulary (irlm-top3K-bad), worsened performance. The results of the irlm-top3K-base-65 run show that medium size dictionaries of the IR language models represented the words in the news sections well. Speech recognition performance did not improve when the 25 *K* dictionaries were enlarged so that they could cover the maximum number of 65 *K* words in this run. As expected, enlarging the number of ranked documents that were selected for the training of the IR language models, in order to minimize the data sparsity problem for LM training, did have a positive effect on ASR performance. Extending this approach by creating language models based on a mixture of a topic specific LM and a general LM (irlm-top3K-news+acues), finally gave the best results, showing some 4 % relative gain in MSWER with respect to a general mixture LM (mixlm-news+acues) that was not adapted to individual topics.

11.4.3 Using improved acoustic models

In a final speech recognition evaluation, the best performing standard LM (Kneser-Ney discounting with interpolation) and mixture LM (topic based mixture LM) were evaluated once again, this time using an improved acoustic model that uses a large recurrent neural net: 1024 hidden units instead of 256 units as in the previous experiment (see also section 5.6.4). In Table 11.7 the results of this evaluation are listed pairwise: first the scores obtained earlier using the small RNN, next the scores that result from using the large RNN.

LM	hidden units	WER	MSWER
kn-ipl	256	36.9%	34.6%
kn-ipl	1024	32.9%	30.9%
irlm-3K-news+acues	256	35.3%	32.5%
irlm-3K-news+acues	1024	31.7%	29.0%

Table 11.7: Comparison of word error rates given two LM configurations and an acoustic model using a small RNN of 256 hidden units and one using a large RNN of 1024 units.

11.4.4 Discussion and conclusions

The experiments described in this section aimed at improving language model performance by focusing on the content of the training data (data quality). Selecting the most appropriate data for the BN task domain in general using the available category labels in the data collection was not successful. Neither removing category clusters from the training data that intuitively did not seem appropriate for the task domain, nor selecting only those clusters that showed an improved perplexity with regard to some domain specific test data when added to the training set, outperformed the approach that simply used all available training data for LM training. It was suggested that data quantity dominates data quality, either because the selected data portions in the clusters are too small for robust LM training, or because the selected data is simply not specific enough to train more robust n-grams conditioned on the task domain. However, the results of the experiments on topic based language modeling (Section 11.4.2) showed that a relatively small amount of training data can be sufficient for LM training provided that the training data closely matches the data in the task domain: using only some 1.75 M words of domain specific training data for the creation of topic LMs (irlm-top5K-base), results in a comparable speech recognition performance as was obtained using the full training data set. However, as will be addressed in more detail below, the significant difference between vocabularies in the respective methods, does not warrant any strong conclusions regarding data quality given these experimental results. Nevertheless, applying a less *ad hoc* clustering approach, for example by using automatic clustering procedures as proposed in Gotoh and Renals (2000) or Seymore and Rosenfeld (1997) may result in more appropriate clustering decisions as obtained using the approach described here, thereby allowing for the training of better language models for the BN task domain in general.

An attempt to improve language model performance by deploying a relatively small amount of data that closely matches the general properties of the task domain, autocues data, was successful. Best results were obtained by applying a mixture framework that merges a general LM trained using a large amount of data with the domain specific LM. These results prove the virtue of having at least some training data available that closely matches the target domain. In practice, it is often not feasible to collect large amounts of data of a certain task domain, as this usually requires the manual generation of transcripts. By applying a mixture framework a large domain mismatch can at least be reduced to some extent.

Applying a mixture LM framework along with a careful selection of training data for a restricted domain that preferably covers one certain topic, can further improve language model performance as was shown in the experiments conducted in section 11.4.2. By using transcripts of an initial speech recognition run as query representation for searching related documents in the data collection for LM training, a topic specific component LM could be constructed that in a mixture framework gave the lowest speech recognition error rates. However, narrowing down the focus domain to the topic of a broadcast news item was enabled by deploying the manually generated topic boundaries. It must be noted that in practice such topic boundaries must be generated automatically which significantly complicates the procedure. Moreover, using a dual pass recognition strategy, one pass for query generation and one for generating the final transcript, and creating both the topic LM and the final mixture LM online, slows down processing time considerably. The experiments show that at the cost of some performance degradation a simple speech recognition configuration could be deployed that takes less time for processing (irlm-top3K-bad). Also using a smaller set of documents for topic LM training quickens the LM generation procedure, again at the cost of some performance. Alternatively, a dual pass strategy can be replaced by an updating strategy that periodically adapts the active language model to the current topic making use of the recognition history. This history in a certain time window could then serve as query representation in a comparable scheme as used in the experiments described here and the active language model can be replaced by a better matching LM given the recognition history. Such an approach is evidently less suitable when topics frequently change as in the broadcast news domain: the active LM may in practice often be behind the current topic. However, in task domains were topics are less fragmented, such as in documentaries, an LM updating scheme may be preferred. Ideally, the effect of applying such a scheme will be that the LM becomes more appropriate when progressing through the data.

One could argue that applying a language model adaptation as was done in this research, actually is more a vocabulary adaptation scheme than a language model adaptation scheme. Although vocabulary selection can be viewed as part of the language modeling procedure, they can equally be viewed as separate processes: one focusing on selecting the most appropriate words given the task domain in order to minimize the out-of-vocabulary rate, the other focusing on capturing the n-gram statistics that are relevant for the task domain in order to obtain accurate word probability estimates given input from the task domain. It is questionable whether the performance improvements obtained in the described experiments are due to improved n-gram statistics. Disturbing effects of improbable n-grams given the task domain, such as in the example of “the horse captures the queen”, may indeed occur when language model training data is not adapted to the content of the task domain, but it may be expected that such an effect is small.

11.5 Summary and conclusion

In this chapter, the subjects discussed in the speech recognition part of this thesis were brought together in a number of speech recognition evaluations. The primary goal was to investigate language modeling in a broadcast news transcription task and a number of language modeling configurations were evaluated.

An optimal baseline trigram language model was obtained in terms of performance and size for the Dutch broadcast news domain, given a substantial newspaper training collection of a few hundreds of millions of words, using the following configuration:

- a vocabulary of the $65\,K$ most frequent words in the training corpus,
- n-gram cutoffs of 2

- Kneser-Ney discounting with interpolation
- and optionally, n-gram pruning with a threshold of 10^{-9}

Attempts to improve the performance of the baseline model focused on data selection and language model merging. It was concluded that:

- the available category labels in the corpus could not successfully be deployed for the improvement of the baseline models. It was suggested to use an automatic clustering procedure as proposed in for example Gotoh and Renals (2000) or Seymore and Rosenfeld (1997) to generate more appropriate clustering decisions for the generation of language models that better suit the task domain.
- using a relatively small amount of autocues data, closely matching the general properties of the task domain, improved speech recognition performance. Best results were obtained in a mixture LM framework, that merges a general LM trained using a large amount of data with a domain specific LM.
- extending the mixture approach by generating mixtures of general language models and topic specific language models, created using IR techniques, was the most successful language model creation procedure tested.

Chapter 12

Speech recognition: Summary and future work

The research and development steps undertaken to reach at a Dutch large vocabulary speech recognition system suitable for application in a spoken document retrieval environment are summarised in this chapter. On the basis of this work, a number of research issues in large vocabulary speech recognition for Dutch, given both the ABBOT system and systems general, and given both the broadcast news domain and other domains, are defined.

Speech recognition summary

In the second part of this thesis, the porting of the *ABBOT* system to Dutch was described and a few research issues that aimed at improving the performance of the Dutch system in a broadcast news transcription task were addressed. Below, the results of the respective research and development steps are summarised in brief.

Training data collection

An important part of the development of a Dutch speech recognition system was dedicated to the collection, preparation and storage of appropriate training corpora both for acoustic model training and language model training. For acoustic model training, two training corpora were created at *TNO Human Factors*[1]. One that contained "journalistic dictation", similar to the Wall Street Journal corpus (WSJ0, Paul and Baker, 1992) used in DARPA's Hub-3 research program, and one containing broadcast news data, comparable with the training data used for the Hub-4 broadcast news evaluations. For language model training, the *PCM Publishers* provided a

[1] See also Appendix C

daily feed of newspaper data of six Dutch newspapers and a number of magazines, resulting in a training corpus, that was fixed for this research to the period January 1999 until December 2001, of 370 *M* words. As the daily feed continued after this period, the newspaper collection is still growing. Furthermore, from the Dutch National Broadcast Foundation (NOS[2]), autocues from broadcast news programs could be obtained. Finally, a teletext capturing card was deployed for collecting teletext subtitling from various news related programs. All text data was carefully normalised and stored in a database.

Although one can argue that the collection of training data is not exactly a research topic, the availability of large corpora for language model and acoustic model training is of significant importance for further improvements in Dutch LVCSR. One of the reasons that English LVCSR systems could obtain the high speech recognition accuracies as reported in the broadcast news benchmark tests, was that the developers of these systems had available huge amounts of training data for system training, provided along with these benchmark tests (see e.g., Graff, 2002). Moreover, for domains other than the broadcast news domain, the collection of domain specific data can be problematic. Mismatches in training and test conditions are then hard to solve, in turn resulting in lower speech recognition accuracies. In the *ECHO* project for example, the only data sources that could be obtained were copies of carbon copies of transcripts of a small subset of the material in the collection. The match with the entire document collection was minimal, but more importantly, using OCR techniques to digitise the paper transcripts for use in language model training failed, due to the low quality of the carbon copies. The development of methods that enable a swift adaptation to various task domains could provide a solution for such problems.

However, to enable the research and application of domain adaptation methods for a certain language, at least a large quantity of training data from various domains, both for acoustic modelling and language modelling, must be available. For the Dutch language, in March 2000, the first release of the “Spoken Dutch corpus” (CGN, Oostdijk, 2000) was published. This corpus currently contains more than 450 hours of orthographically transcribed speech and will be augmented to some 1000 hours of speech in the future. As the preparation of such a large corpus for training is laborious, it was decided not to use the CGN corpus for development and research purposes in the context of this thesis, but deploy it in future research as described below. For language model training, the CGN corpus is less suitable, as this typically requires several hundreds of millions of words of text data. With this research, the collection of a large Dutch newspaper corpus (Twente Nieuws Corpus) was initiated, in July 2003 containing about 450 *M* words of text data. Although the collection of newspaper data will continue, to enable language model training for other domains then

[2] see Appendix C.1

the news domain, exploring other text data sources is required. Moreover, a shortcoming of newspaper data in the context of language modelling for speech recognition is that is consists of written text. To improve language model performance for spontaneous speech, a reasonably large collection of speech transcripts, such as the broadcast news transcripts that are available for English, would be helpful.

Related to large audio and text collections for ASR training purposes, is the need for robust tools for a flexible processing of these amounts of data, such as databases, and a wide range of auxiliary tools, such as normalisation tools. Adaptation techniques in particular, such as the vocabulary adaptation techniques described in Chapter 9, require that available data can be accessed easily. Data management is however sparsely discussed in the context of LVCSR. Although perhaps not a research issue in itself, having flexible and robust tools available within a solid data architecture is an important prerequisite for further improvements in LVCSR.

Text normalisation

A relatively large amount of effort was spent on text normalisation. Next to a number of obvious normalisation procedures, a number of specific algorithms were devised to normalise the text data collection as good as possible. Although it can be assumed that applying a less detailed normalisation procedure would sparsely have an effect on the eventual speech recognition performance, it was interesting to note that it appeared to be so difficult to prevent that ’strange’ lexical items showed up in the vocabularies and language models. Often such lexical items were not accounted for by the normalisation routines, sometimes they emerged from the normalisation routines themselves.

The normalisation procedures resulted in a 64 % decrease in distinct words and managed to reduce the lexical variability in the text data substantially. Of the normalisation procedures that focused on variant reduction, especially the case normalisation step substantially reduced lexical variability. The effect of the error correction normalisation steps (spelling and diacritics) was much smaller, but at least a relatively large number of misspelled words could be corrected.

Word pronunciations

As predicting accurately which words are to be expected in the broadcast news domain is difficult, typically large vocabularies are deployed for speech recognition in this domain. Usually not all word pronunciations for the words in these vocabularies are available in a background lexicon, so a grapheme-to-phoneme (G2P) converter is an indispensable tool. The development of a G2P tool was described that applies a learning algorithm and uses a decision tree for the generation of pronunciations. This G2P achieved a reasonable pronunciation generation accuracy of 90 % for unseen words. Along with a large background pronunciation lexicon, the G2P became a valuable tool in further development and research steps.

Although the G2P achieved a reasonable pronunciation generation accuracy of 90 % for unseen words, it was explained that the training procedure, relying on a large proportion of rewriting rules and null-insertion rules, is open to improvement. Circumventing the manual generation of the majority of these rewriting rules deploying a dynamic programming algorithm (Kienappel and Kneser, 2001; Mana et al., 2001), is currently being investigated.

Besides improving the training procedure of the G2P, it can be worthwhile to investigate whether pronunciation variation can be incorporated. It is well-known that pronunciation variation is a source of error in speech recognition and a number of approaches have been proposed (such as those of Kessens, 2002; Wester, 2002, for Dutch) to deal with this problem. In automatic grapheme-to-phoneme conversion, pronunciation variation is usually not addressed explicitly, as the goal is merely to enable the generation of a normative pronunciation of a word when the pronunciation cannot be derived from an, often carefully constructed, background lexicon. However, instead of regarding a G2P tool as an auxiliary tool that is only deployed in special circumstances, a G2P can also be viewed as a speech recognition lexicon itself, dynamically providing the word pronunciations that are needed at a particular stage in the recognition process. In the ideal case, the G2P provides those pronunciation variants that are most likely given some *general* knowledge that it has obtained earlier about the pronunciation of a given word, and some *task-conditioned* knowledge about the pronunciation of words, for example generated on the basis of a recognition history. The general knowledge could for example be obtained using a data-driven approach that for example deploys forced alignment techniques and decision trees for collecting pronunciation variation statistics from automatic transcriptions of a relatively large, general speech corpus, such as for Dutch the “Spoken Dutch Corpus”. This general pronunciation knowledge could be represented using word pronunciation probabilities. The task-conditioned knowledge may then be used in two ways: firstly, to weight the probability distribution according to the local context and secondly, to adapt the general knowledge given the local pronunciation observations, resembling a continuous learning process. An example of deploying weights using task-conditioned pronunciation knowledge, could be assigning more weight to pronunciations containing certain phone deletions given that these were frequently observed in a task’s history. Although the implementation of such an approach may be complicated and undoubtedly introduces new problems (setting thresholds, incorporating phonological/phonetic knowledge), it may provide a framework for a dynamic handling of pronunciation variation in large vocabulary speech recognition tasks.

In general, the handling of the, frequently occurring loanwords and foreign names in the BN domain, deserves some more attention in G2P development. As the pronunciation of these words often contradicts Dutch pronunciation rules, the correct word pronunciations will often not be generated by a G2P, even when these words had been included in the training set. Currently, the only solution seems to include loan words and foreign names as often as possible in a background lexicon. Preferably, one would deploy some sort of language detection and generate a pronunciation on the basis of the language classification. For example, when the language detection tool classifies a word as being an English word, an English G2P could be consulted for the generation of the pronunciation.

A final issue that needs to be addressed in the context of future research in grapheme-to-phoneme conversion is compound splitting. The chance that a Dutch compound word is not in the background lexicon, and hence, its pronunciation has to be obtained via a G2P tool, is generally higher than for non-compound words as new compounds can be easily invented. However, by splitting the compound into its components, a pronunciation could still be generated from the lexicon by concatenating the available pronunciations of the components. In order to produce correct pronunciations, co-articulation rules must be applied during the concatenation process. Although compound splitting was addressed in this thesis in the context of vocabulary construction, it was not yet applied within the context of word pronunciation generation. It is worthwhile investigating how much the amount of word pronunciations that can be provided by the background lexicon (currently some 75 %), can be improved by applying compound splitting, and a procedure for component concatenation and co-articulation correction.

Acoustic modelling

The acoustic models for this research were based on a relatively small broadcast news corpus containing 14 hours of speech, that was especially created for this research. In spite of the small amount of training data compared to the amounts that are typically used for the Hub-4 evaluations, the acoustic models provided a reasonable performance in terms of phone error rate on the broadcast news test data, especially when model merging was applied. By merging the output of the RNN acoustic models trained forwards and backwards in time, phone error rates could be improved with 4–5 % relative.

Regarding the acoustic modelling part of a LVCSR system, a number of research topics can be identified. An interesting topic is how the CGN corpus can best be deployed to improve LVCSR performance in the BN domain and in general. The corpus could for example be used for the training separate acoustic models for specific acoustic conditions, especially the training of gender and bandwidth dependent models. With the relatively small amount of available training data gender and bandwidth dependent modelling was not considered for this research. As the speech data in the corpus is collected from a variety of domains, the question is how this data can best be exploited for a given task domain. A possible approach is to divide the data into a number of global domains. Domains that have a certain resemblance with the target domain can then be added to a training set. Alternatively, different models could be trained and merged at run-time to obtain weighted phone probability estimates based on different information sources. Furthermore, this corpus enables the investigation of acoustic model adaptation techniques. Other research topics include the application of lightly supervised or unsupervised training (Lamel et al., 2001), and the automatic identification of speaking styles, accents and non-native speech.

Vocabulary construction

The selection of words for the language model vocabulary was regarded as an important issue in the context of a Dutch broadcast news transcription task, in order to reduce the number of out-of-vocabulary words for this domain. Word selection was addressed as being a matter of appropriate training data partitioning: either on the basis of content information or on the basis of temporal information. Using temporal information for data partitioning was discussed in more detail. A number of vocabulary selection methods based on temporal data partitioning and word frequency information were discussed. It was argued that word frequency information alone is inadequate to predict which words are to be expected in a particular news show at a specific point in time. In order to capture word importance dynamics in a domain that typically shows large word fluctuations with this respect, a novel method was introduced, referred to as the binary prediction method. This method tries to incorporate temporal information directly into the selection procedure. Indeed, this method gave the best OOV performance in a vocabulary selection experiment, that compared a number of different vocabulary selection techniques. However, the gain was too small to warrant strong conclusions regarding this method.

Speech recognition performance could benefit greatly from an accurate prediction of the words that are most likely to appear at a certain point in a speech recognition task. It would reduce the OOV rate and in the ideal case, prevent the selection of words in the vocabulary that have a low chance of occurring, enabling the use of smaller vocabularies. In this research, temporal information was deployed in a highly simplified way, as a first approach to improve word prediction in the broadcast news domain. Although only small lexical coverage improvements were obtained, the results showed that temporal information can provide additional information about word occurrence. But still, large 65 K vocabularies had to be deployed to obtain a comparable lexical coverages as were obtained using the standard relative frequency approach. Further research should aim at improving the representation of temporal information so that it can more adequately be used for word prediction.

Compound splitting

That compound splitting could improve speech recognition performance for Dutch has often been suggested but was not yet fully investigated in a full scale, Dutch speech recognition experiment. For this purpose, a data-driven compound splitting algorithm was created that used an alphabetically sorted word frequency files based on a large amount of newspaper data. Although recall of the algorithm could not be computed, it was explained that the design of the algorithm guarantees that compound recall will at least be sufficiently high for the purpose of the experiment. Precision of the algorithm, measured on the first iteration of the algorithm and using a commercial dictionary with constituent boundary labels, was 97.7 %.

A first comparison between the original text version and a fully decomposed text version showed a 20 % reduction in distinct words and a better lexical coverage for lexicons derived from the decomposed data. Restricting the compound splitting procedure aiming at even better lexical coverage performance was investigated in detail by looking at the contribution to the self-coverage of a 65K lexicon of individual decompositions. When such individual decompositions did not improve lexical coverage by themselves, they were excluded from the list of compounds used for the actual decomposition of the text data. However, this procedure did not yield a robust improvement of lexical coverage of lexicons.

A number of language models were created to investigate whether compound splitting also improves speech recognition performance. One language model was based on the original text data (baseline) and the others were based on text data that was decomposed in different ways: two using a full compound splitting procedure without applying any restrictions, and three using a restricted compound splitting procedure that only decomposed compounds when they were not frequent enough to be included in a lexicon of size N, where N was chosen to be 5 K, 20 K and 65 K. For the unrestricted procedure, two text version were created: one that treated the binding morpheme as a separate constituent and one that attached the binding morpheme to the preceding constituent. The language models were evaluated in a broadcast news transcription task. The language models that were created using the restricted compound splitting procedure, in which compounds in the 0-20 K word frequency range were omitted, gave the best speech recognition performance. However, the performance difference with the language models based on unrestricted compound splitting was only marginal (1 % absolute).

Although the research described in this chapter confirmed the hypothesis that compound splitting can improve lexical coverage and speech recognition performance for Dutch, some issues remain that deserve to be looked into more closely. Firstly, for specific tasks, having available an accurate reverse compound splitting procedure (that maps constituents back to the original compounds as addressed in Section 10.5.1) may be crucial. In a spoken document retrieval framework, reverse compound splitting is of minor importance, but in a dictation task for example, it may be expected that a user of a speech recognition system requires compounds to be reassembled automatically. Applying reverse compound splitting in this research introduced a substantial amount of incorrect compounds. In order to improve this procedure, the compound detection algorithm could be improved so that incorrect compound mappings can be prevented. Alternatively, the compound splitting table can be post-processed to exclude entries that introduce incorrect mappings.

A second issue for future research is the exact behaviour of compounds and decomposed compounds in the language model. Given that WER improved with increasing OOV rates using language models based on the restricted splitting procedures, it was hypothesised that this procedure enables the creation of more robust n-gram language models. However, this research could not provide evidence for this hypothesis. It may therefore be worthwhile to search for an experimental design that enables the investigation of compound splitting on the language model level. Furthermore, the effects of applying the opposite of compound splitting could be included in this investigation: the combination of frequent orthographic word tuples, referred to as multi-words, into single items in the recognition lexicon, as proposed for example by Gauvain et al. (1997).

Language modelling

Most of the subjects discussed in the speech recognition part of this thesis were practically brought together in a number of speech recognition evaluations. The primary goal in these evaluations was to investigate language modelling in a broadcast news transcription task and a number of language modelling configurations were evaluated. An optimal baseline trigram language model was obtained, in terms of performance and size for the Dutch broadcast news domain, and given a substantial newspaper training collection of a few hundreds of millions of words, using a vocabulary of the 65 K most frequent words in the training corpus, n-gram cutoffs of 2, Kneser-Ney discounting with interpolation, and optionally, n-gram pruning with a threshold of 10^{-9}. Attempts to improve the performance of the baseline model focused on data selection and language model merging. It was concluded that the available category labels in the newspaper text corpus could not successfully be deployed for the improvement of the baseline models. It was suggested to use an automatic clustering procedure as proposed in for example Gotoh and Renals (2000) or Seymore and Rosenfeld (1997) to generate more appropriate clustering decisions for the generation of language models that better suit the task domain. Moreover, using a relatively small amount of autocues data, closely matching the general properties of the task domain, improved speech recognition performance. Best results were obtained in a mixture LM framework, that merges a general LM trained using a large amount of data with a domain specific LM. Finally, extending the mixture approach by generating mixtures of general language models and topic specific language models, created using IR techniques, was the most successful language model creation procedure tested.

It was already mentioned at the start of this section that in order to extend the focus of language models beyond the broadcast news domain, extending the coverage of the LM training data is required as well, as language models are very sensitive to training and test mismatches. In this research domain adaptation was briefly investigated by deploying a dual-pass decoding strategy along with information retrieval techniques. However, as processing time slowed down considerably applying this method, optimising this procedure or even identifying other techniques is required.

Evaluation

In real-life speech recognition applications, the task domain is dynamic which requires that a speech recognition system that was once developed for a domain is adapted and evaluated frequently. Monitoring and evaluating a speech recognition system however, requires a speech recognition expert and evaluation data. For many ASR applications, for example a system that is used for the daily recognition of broadcast news for a SDR application, the frequent deployment of a speech recognition expert and the generation of evaluation data is costly. To provide for a manageable monitoring and evaluation procedure for this type of systems, (semi-)automatic evaluation mechanisms are indispensable.

Porting *ABBOT* to Dutch

The results of the speech recognition evaluations described in Chapter 11, confirms that the *ABBOT* system has been successfully ported to Dutch. A baseline performance in the broadcast news domain of 34.6 % MSWER was obtained and the best performance was achieved using large RNNs for the acoustic models and a multi-pass decoding strategy along with information retrieval techniques for the creation of optimal language models, yielding a MSWER of 29.0 %. Although this performance cannot equally be compared with the performances of English systems participating in TREC (between 20 % and 30 % WER) due to the different experimental conditions, it can be concluded that a major step forward in catching up with the international state-of-the-art in broadcast news transcription is achieved. At least, the performance of the Dutch system can be regarded as satisfactory for the envisaged spoken document retrieval task, given the observations in the TREC SDR tracks that speech recognition with a word accuracy of about 60 % can already successfully be deployed for spoken document retrieval

(Garofolo et al., 2000). Possible research and development directions aiming at bridging the final gap between the performance of the current Dutch system and English state-of-the-art systems, are discussed in the next section.

Part III

Spoken Document Retrieval

Chapter 13

An illustrative SDR experiment

This chapter gives an illustrative example of the application of the Dutch speech recognition as described in Part II of this thesis in a spoken document retrieval task in the broadcast news domain.

13.1 Introduction

The ultimate goal of the research and development steps described in the preceding part, was to obtain a speech recognition system that could successfully be deployed for Dutch spoken document retrieval tasks. It was concluded in Chapter 12 that the *ABBOT* system had successfully been ported to Dutch and that the system is ready to be deployed in a spoken document retrieval framework, either in a *LVCSR configuration* which has been the main focus of the research in this thesis, or alternatively, in a *keyword spotting* configuration or a configuration based on *sub word units* (phones), as described in detail in Chapter 2.

With the TREC spoken document retrieval tracks, ample experience has been obtained with the evaluation of spoken document retrieval systems. In the first SDR evaluations performed during TREC-6 in 1997, a *known-item* retrieval task was chosen for evaluation. A known-item retrieval task simulates a user seeking a particular, half-remembered document in a collection. The SDR system in such a task, is required to generate a single correct document for each query, rather than a set of documents ranked according to relevance, as in an *ad-hoc* task (Voorhees et al., 1997). The ad-hoc task was used for evaluation from TREC-7 onward. Here, participating systems provide for every query (called “topic” in TREC) in the task a list of documents that are ranked according to relevance. The top 100 documents that were retrieved by participating systems and by judges performing manual and semi-automatic searches, referred to as the *pool*, were then assessed by "judges" providing relevance judgements: whether a document in the pool is relevant given a query or not.

For an ad-hoc style SDR evaluation not only a realistically large test collection is required, but also judgements need to be generated. Given the available time and resources, such an evaluation was not feasible. Therefore, a *known-item* retrieval task was chosen in order to be able to demonstrate the application of Dutch speech recognition in a first Dutch SDR evaluation.

13.2 Experimental design

For the known-item SDR evaluation, a set of 18 television news broadcasts ("*NOS Acht uur journaal*") from January 2002 until March 2002 were collected and manually transcribed on the word level. Manually segmented (hand-annotated temporal story boundaries were given): 180 stories, mean length of 257 words. Introductions and weather reports were excluded. Story topics were generated by students who were instructed to create topic "titles" that in a few words (with a maximum of ten words) give a reasonable impression of the contents of the story. These titles were further interpreted as query aiming at the retrieval of the respective story. The retrieval task was to find for every query, the target story. The titles were used as queries for retrieval given the following evaluation modes:

- using document representations that are based upon perfect, human-transcribed reference.

- using document representations based upon a speech recognition system producing a relatively large number of errors. A speech recognition configuration with a 5 K vocabulary and language model that had obtained a mean story word error rate (MSWER) of 50.9 % was used (see Chapter 11).

- representation based upon a relatively well performing speech recognition system. The best performing baseline system that used Kneser-Ney discounting, a 65 K vocabulary and a large RNN (1024 hidden units) produced the transcripts. This system obtained a MSWER of 30.9 % on the broadcast news test data described in Chapter 11.

- using a phone-based document representation. Here the speech recognition system was used as a phone recogniser with a large RNN trained on the TNO-BN corpus as described in Chapter 5. Two types of document representations were created using the phone outputs: one uses sub word units of three phones with N phone overlap, the other uses four phones with N phones overlap.

For retrieval, Okapi term weighting was applied as described in Section 11.2. The following evaluation methods as applied at TREC-6 were used:

- *Mean rank when found* (MRWF), defined as the mean rank at which the target story was found, averaged across all queries that retrieved the target stories in all retrieved documents.
- *Mean reciprocal rank* (MRR), defined as the mean of the reciprocal of the rank at which the target story was found over all queries, using 0 as the reciprocal for queries that did not retrieve the story.

13.3 Results

Table 13.1 shows the results of the known-item retrieval task. Using the reference transcript as document representation gave the best retrieval performance in terms of found documents and mean reciprocal rank. Using high quality speech recognition produced comparable results: only one more document was not found (10 instead of 9) and the mean rank when found was even slightly better compared to one obtained in the reference condition. As could be expected, deploying a low quality speech recognition system significantly worsens retrieval performance. Almost a quarter of the documents could not be found, on average, the target stories were retrieved almost one rank lower, and the mean reciprocal rank decreased almost 35 % relative compared to the high quality speech recognition condition. In the sub-word based conditions only half of the queried documents could be found and in such cases, the target documents appeared on average at rank 8. Mean reciprocal rank dropped compared to the ASR-high condition with almost 44 % relative.

document representation	MRWF	MRR	not found
Reference	2.0778	0.7462	9 (5 %)
ASR-high	1.9278	0.7689	10 (5.6 %)
ASR-low	2.8556	0.5061	43 (23.9 %)
3-phone	8.3042	0.3319	87 (48.3 %)
4-phone	8.1939	0.3278	87 (48.3 %)

Table 13.1: Mean rank when found (MRWF), mean reciprocal rank (MRR) an number of documents not found given 180 queries for document representations based upon the reference transcript, speech recognition with a low performance (ASR-low), speech recognition with a good performance (ASR-high) and phone recognition with 3-phones and 4-phones with an overlap of N phones.

13.4 Conclusions and future research

The known-item retrieval task illustrated well how speech recognition transcripts can be used for the retrieval of Dutch broadcast news programs and that the speech recognition system described in this thesis, can successfully be deployed in a spoken document retrieval task. Speech recognition performance was significant in this experiment. Retrieval performance deteriorated when LVCSR performance dropped to some 50% MSWER. The results of the subword unit based approach, deploying a phone recogniser, were disappointing. Its retrieval score and the difference in performance with the LVCSR systems, does not justify further research into this area. Further research is needed to investigate Dutch SDR using the current Dutch speech recognition implementation in a more realistic ad-hoc style experimental set-up using a larger test collection.

Chapter 14

Summary and conclusions

With a reference to the original goals of this thesis, this chapter summarises the research and development work that was addressed, provides an overview of the conclusions that were based upon this work, and discusses possible future research directions.

14.1 Original goals of this thesis

The goal of the research and development work described in this thesis was to realise a conceptual and practical framework for the investigation of a wide range of issues related to information retrieval in the context of multimedia and spoken-word collections. The main focus in this thesis is on solving the representation mismatch between natural language queries (text) and the representation of documents in such collections (audio and/or video) using speech recognition techniques, referred to as spoken document retrieval (SDR). International publications on SDR have shown that deploying a large vocabulary speaker-independent continuous speech recognition (LVCSR) system is generally the best option for SDR. As a Dutch LVCSR system was not available, an important goal was to develop and implement a Dutch system that could be used in a Dutch SDR framework, and to set a baseline LVCSR system to enable further research in the field of Dutch LVCSR. Using this baseline system, the aim was to contribute to Dutch LVCSR research by addressing issues in Dutch LVCSR that had received almost no attention: compound splitting, vocabulary selection and language modelling. Finally, the goal was to demonstrate the applicability of the Dutch system in an experimental retrieval environment.

14.2 Summary

The first part of this thesis (Part I), addresses the theoretical framework of information retrieval in the context of multimedia and spoken-word collections. It is explained how the query/document representation mismatch in these collections can be solved by using SDR and a number of speech recognition techniques that have been deployed in this field are discussed. Part II, provides a detailed specification of the development of the Dutch LVCSR system. First the characteristics of the English *ABBOT* speech recognition system, a hybrid RNN/HMM system that was used as a starting point for the Dutch system, are described. Next, the generation of word pronunciations for the speech recognition dictionary is addressed. It is shown that given the large vocabularies and the characteristics of Dutch (i.e., word compounding), a tool for automatic pronunciation generation (automatic grapheme-to-phoneme conversion) is indispensable. As ready-to-use grapheme-to-phoneme (G2P) conversion software was not available for Dutch, a Dutch G2P tool was developed which is also described in the second part. The remainder of this part consists of a specification of the training of the acoustic models and language models suitable for speech recognition in the broadcast news domain, along with a broadcast news LVCSR performance evaluation. Specific topics that are addressed are the collection and preparation (normalisation) of speech corpora and text corpora for training, optimisation of the speech recognition vocabulary using word selection methods and improving speech recognition performance by applying compound splitting. Part III finally demonstrates the application of the Dutch LVCSR system in SDR by describing the results of a known-item retrieval experiment using a collection of broadcast news programs.

14.3 Overview of conclusions

By providing an overview of information retrieval in the context of multimedia and spoken-word collections and a detailed listing of the main issues in spoken document retrieval, Part I contributes to the realisation of a conceptual framework that can further be explored and refined. The prototype Dutch SDR set-up can be further improved and extended to other domains and applications. Given the observations in the TREC SDR tracks that speech recognition with a word accuracy of about 60 % can already successfully be deployed for spoken document retrieval, the system can be regarded as satisfactory for spoken document retrieval tasks in general, as a performance in the broadcast news domain of on average between 30 % and 40 % WER was achieved. However, as spoken document retrieval performance benefits from an improved speech recognition performance, directing further research to speech recognition performance optimisation remains useful. Moreover, as for other task domains than broadcast news, achieving a comparable speech recognition performance can be complicated (for instance in less structured domains, or domains for which language model training corpora are difficult to obtain), further research is necessary. The developed baseline Dutch LVCSR system can appropriately be deployed for this.

In this research, the baseline LVCSR system was used to investigate a number of research topics that are especially relevant for the development of a Dutch LVCSR system. It was shown that due to this high lexical variability in Dutch, it was shown that either a very large background lexicon and/or a robust grapheme-to-phoneme converter are indispensable for the generation of word pronunciations for the recognition dictionary. This is especially the case in dynamic task domains such as the broadcast news domain that requires frequent vocabulary updating. Complications of the high lexical variability in Dutch also showed up when words had to be selected for the speech recognition vocabulary. It is explained that a high lexical variability means that vocabulary space is sparse, requiring a careful selection of words for the speech recognition vocabulary to reduce the number of out-of-vocabulary words. After investigating a number of word selection mechanisms, the following conclusions were drawn:

- A selection of approximately one year of recent newspaper data may be regarded as an optimal starting point for a word frequency based selection of vocabulary words for the representation of Dutch news broadcasts. One year is long enough to capture words that have a not very high but consistent frequency, and short enough to reduce the chance of unintentionally including words in the vocabulary that have built up high frequency counts in the past but are not relevant anymore in up-to-date news events.

- In longitudinal speech recognition tasks in the broadcast news domain, periodic updating of the vocabulary is necessary as to enable the recognition of new words that gradually appear over time.

- A simple but effective procedure for periodic updating is the use of a shifting look-back time-window of approximately one year for word selection based on word frequency counts.

- The results of the experiment suggest that including temporal word usage information in the word selection procedure can improve OOV performance of vocabularies, provided that the word frequency information is already robust in itself.

Further research is needed to search for a better paradigm for representing temporal information. In this research, temporal information is used in a highly simplified way and in a compressed format. It can be argued that by doing so, only little temporal information is helpful in the vocabulary selection procedure.

The investigation of compound splitting in the context of LVCSR stems from the lexical variability considerations in Dutch. It was investigated whether compound splitting could effectively be deployed to enlarge the coverage of speech recognition vocabularies so that OOV words can be reduced. The following observations are worthwhile summarising:

- Using a large text corpus and a purely data-driven compound splitting algorithm, a compound splitting table can be generated with a high recall and precision.
- When there are multiple splitting alternatives, the most plausible splitting alternative can often be found by using the frequency of the compound parts in the training data and the within-compound frequency of the parts.
- Compound splitting improves the lexical coverage of large vocabularies selected on the basis of relative frequencies in the training data. Excluding compounds from the splitting procedure that may not improve lexical coverage when split given their relative frequency ranking, does not further improve lexical coverage.
- Language models based on decomposed text data improve speech recognition performance. Best speech recognition results are obtained when compound words in the 0-20 K word frequency range are not decomposed. This indicates that highly frequent compound words are already adequately modelled.

Further research directed toward compound splitting in Dutch LVCSR must provide more insight into the behaviour of compound words either decomposed or not, in the language model. Moreover, as in this research compounds were decomposed only when their constituents had a minimum length of six characters, it is worthwhile to investigate whether an optimal constituent length can be found experimentally.

To set a baseline for Dutch language modelling, a number of n-gram language modelling schemes were evaluated with regard to Dutch speech recognition performance. On the basis of these evaluations it was concluded that:

- Due to the decrease in OOV words, larger speech recognition vocabularies up to 65 K words give better speech recognition performances in spite of an increased acoustic confusability.
- The effect on speech recognition performance of choosing a particular smoothing technique is marginal for the Dutch broadcast news domain and given newspaper training data.
- Interpolating a general language model based on a large collection of (newspaper) data and a domain specific language model based on domain specific (autocues) training data significantly improves speech recognition performance.

- Narrowing down the focus of the language model creation procedure to the topic-level, by generating mixtures of general language models and topic specific language models using a dual-pass decoding strategy and IR techniques, the quality of a language model can be further improved at the cost of a substantial increase of processing time.

Regarding the development of the acoustic models for the *ABBOT* system it was concluded that a relatively small amount of 14 hours of domain specific acoustic training data is already sufficient to obtain a remarkably good speech recognition performance using the hybrid RNN/HMM framework of the *ABBOT* system. It may be expected that adding more training data in the order of magnitudes that are customary in the HUB4 evaluations, and by training context-dependent acoustic models (gender, bandwidth), the current speech recognition performance can further be improved.

Along with the development of a baseline Dutch LVCSR system, the following contributions were made to Dutch LVCSR research:

- *Newspaper corpus* (TwNC)
  The collected Dutch newspaper data were converted to a uniform XML-format and made available to the research community as the *Twente News Corpus* (TwNC). The collection contains 370 *M* words of data. A number of research sites currently use the corpus for a variety of research activities.

- *Normalisation module*
  Along with the newspaper corpus, the source-code (in Perl) of the developed normalisation routines is provided.

- *Grapheme-to-phoneme converter* (G2P)
  A version of the G2P that was trained using the CELEX lexical database was made available for UNIX and Windows.

14.4 Future directions

A number of possible future research topics regarding Dutch LVCSR and SDR have already been mentioned in this chapter and previous chapters. Obviously, the list of topics is far from being exhaustive and additional research topics can be thought of that may in one way or another improve LVCSR and/or SDR performance for Dutch. To keep up with the international technological advances and state-of-the-art, and to enable participation in future collective research initiatives on an international level in the domain of spoken-word audio collections, a number of research and development topics require priority attention.

14.4.1 Training corpora

For further speech recognition improvements, additional training corpora, especially for language model training need to be collected. For acoustic model training, the CGN corpus is very well suited for further Dutch LVCSR research, although it must be noted that this corpus contains speech from a variety of domains and may not be very suitable for domain specific acoustic model training requiring large amounts of in-domain data. The development of tools for unsupervised acoustic model training (see e.g., Lamel et al., 2001) might be suited when an SDR application has to be extended to other domains than the broadcast news domain. For language model training, only newspaper data is widely available in sufficient amounts. However, the resemblance of this type of data with real life speech is known to be low. The collection of text corpora that better match speech in real life applications needs special attention. Alternatively, research could focus on modelling spontaneous speech using the available written-text corpora, for example, by incorporating breath noises and hesitations at specific points in the data. At least, investigating the extent to which written-text corpora do or do not match real-life speech in Dutch, is an interesting starting point for future research.

14.4.2 Domain specific models

Improving Dutch LVCSR performance to a level that is comparable with the international state-of-the art, is not simply a matter of addressing specific research and development topics. In general, each topic may contribute a few percentages, or tenths of percentages, to the overall improvement of a system. All contributions together may eventually result in a substantial improvement. However, at least for the broadcast news domain, gender-dependent and bandwidth-dependent acoustic modelling have proven to boost speech recognition performance significantly. This type of acoustic modelling may therefore be a good choice to start with to improve the current Dutch LVCSR system. Apart from better matching training corpora, language modelling can benefit from research aiming at a more accurate prediction of the content (speech type, topic, named entities) of specific parts in the task domain. Accurate content prediction enables the selection of high quality training data, the creation of topic specific (mixture) language models, and the creation of vocabularies with a better coverage.

14.4.3 Maintenance and monitoring

Other topics that need to be addressed are related to the inclusion of LVCSR techniques into realistic SDR applications. An important consideration for such applications is that they need to be maintained and monitored. In most realistic task domains, speech recognition components need to be adapted to changes in the domain with a certain frequency. It may for instance be desirable to adapt available speaker-dependent acoustic models to a new television news anchor-man/woman, update the language model, or check the automatically generated word pronunciations (especially of named entities) that have recently entered the domain. Moreover, especially in SDR applications, speech recognition performance degradations cannot easily be noted. Regularly monitoring the system's performance is therefore desirable. Making adaptations by "hand" using an expert, or letting an expert perform system evaluations, is not only costly, but often impractical as well. Investigating whether maintenance and monitoring can be performed (semi-)automatically or lightly supervised, is becoming increasingly important since SDR performance is good enough to be implemented in realistic applications.

In general, having better means available to evaluate Dutch LVCSR and SDR is an important prerequisite for further research and development. As evaluation corpora and test collections are costly to develop, often only relatively small amounts of evaluation data can be used. In some cases, certain evaluation methods cannot be used at all. For the evaluation of Dutch SDR for example, preferably a TREC-like SDR corpus along with relevance judgements should be available. However, such a collection was not within reach for this research. Initiatives aiming at improving the means for Dutch LVCSR and SDR evaluation will therefore be welcomed.

Appendix A

Short description of UT projects involving ASR

A.1 OLIVE

The goal in the OLIVE project was the development of a multilingual indexing tool for broadcast material based on speech recognition. The Olive project was funded by the European Commission under the Telematics Application Program. It started in 1998 and lasted until 2000. Speech recognition for German and French was developed by LIMSI and Vecsys (France).

A.2 DRUID

DRUID aimed at the development of tools for the indexing and retrieval of multimedia content on the the basis of image processing and language and speech technology’. It was a project within the scientific program of the Telematica Instituut, an institute is co-funded by the Dutch government and a number of industrial partners. DRUID started in 1998 and lasted until the end of 2001.
DRUID project website: `http://dis.tpd.tno.nl/druid/`

A.3 ECHO

The ECHO project developed a software infrastructure to support digital film archives, to provide web-based access to collections of historical documentary films of great international value and to increase the productivity and cost effectiveness of producing digital film archives (see also, Ordelman, 2000) . In this project, speech recognition was one of the tools for content disclosure. ECHO was a project within the 5th Framework Program IST (Information Society Technologies) of the European Union. It started in 2000 and ended in 2003.
ECHO project website: `http://pc-erato2.iei.pi.cnr.it/echo/`

A.4 MUMIS

In the MUMIS, basic technology for automatic indexing of multimedia program material was developed. Domain of focus was soccer and speech recognition was one of the technologies applied. MUMIS was a project within the 5th Framework Program IST (Information Society Technologies) of the European Union and was funded under the 5th Framework Program of the European Community). MUMIS started in 2000 and ended in 2003.
MUMIS project website: `http://parlevink.cs.utwente.nl/projects/mumis/`

A.5 WATERLAND

The WATERLAND project focuses on semi-automatic techniques for metadata extraction in the digital production of media. Improvement of the existing speech recognition technology for Dutch, is one of the aims of the project. Furthermore, the evaluation of system components is addressed specifically. The project started in June 2001 and will last until the end of May 2005.
WATERLAND project website: `http://www.innovatie.nob.nl/waterland/`

Appendix B

DRUID phone set

IPA	DRUID	EXAMPLE	IPA	DRUID	EXAMPLE
			p	p	pak
	sil		b	b	bak
a	a:	naam	d	d	dak
e	e:	heet	t	t	tak
o	o:	boot	k	k	kat
u	u	boek	g	g	goal
y	y	vuur	f	f	fel
i	i	riet	v	v	vel
ə	@	gemak	s	s	set
ɑ	A	bak	z	z	zet
ɛ	E	pet	x	x	toch
ɪ	I	pit	h	h	hand
ɔ	O	pot	ʃ	S	sjaal
œ	U	put	ʒ	Z	jam
au	AU	goud	m	m	man
ɛi	EI	fijn	n	n	naam
ø	EU	reus	ŋ	N	bang
ʌy	UI	huis	l	l	land
øɹ	EUr	deur	r	r	rand
eɹ	e:r	beer	w	w	wit
oɹ	o:r	boor	j	j	jong
			t	tj	watje

Table B.1: DRUID phone set

Appendix C

List of Names, Institutions and Software

C.1 Names and institutions

- Van Dale

  Van Dale Lexicografie is one of the major dictionary publishers in The Netherlands and provided the pronunciation lexicon that was used in this research.

  URL:`http://www.vandale.nl`

- PCM Publishers

  Dutch newspaper publisher that provided the newspaper data that were used in this research of the following Dutch newspapers: Volkskrant, NRC Handelsblad, Trouw, Algemeen Dagblad en Het Parool.

  URL:`http://www.pcmuitgevers.nl`

- NIST

  US National Institute of Standards and Technology. NIST's mission is to develop and promote measurement, standards, and technology. The TREC (`http://trec.nist.gov`) Conference series is co-sponsored by the NIST, Information Technology Laboratory's (ITL) Retrieval Group.

  URL:`http://www.nist.gov`

- LDC

  Linguistic Data Consortium, supports language-related education, research and technology development by creating and sharing linguistic resources: data, tools and standards.

  URL:`http://www.ldc.upenn.edu/`

- ELRA

  The missions of the European Language Resources Association are to promote language resources for the Human Language Technology (HLT) sector, and to evaluate language engineering technologies.

  URL: `http://www.elra.info/`

- SPEX

  The Speech Processing EXpertise centre was founded in 1987 with the objective to develop and provide software, tools and databases for companies and institutes active in research and development in the general field of speech, with an emphasis on speech technology. SPEX has a special task in making available spoken language resources for research purposes in the Dutch academic environment.

  URL:`http://www.spex.nl`

- Beeld&Geluid

  The Netherlands Institute for Sound and Vision was established in 1997 as the result of a merger between three large audiovisual archives and the Broadcast Museum. Substantial collections of radio and television programmes, documentaries, commercials, amateur films, photographs and music are all to be found at this institute. The (total) archival holdings include materials dating from the earliest days of cinema right up to current news broadcasts. Estimations of the size of the archive range from 800.000 to almost one million hours worth of viewing and listening. The major collection currently held is that of the (Dutch) public broadcasters, the magnitude of which increases daily.

  URL:`http://www.beeldengeluid.nl`

- NOS *Nederlandse Omroep Stichting* (Dutch National Broadcast Foundation).

  URL:`http://www.omroep.nl/nos/noshome/index.html`

C.2 Software

- *HTK-toolkit*

  The Hidden Markov Model Toolkit (HTK) is a toolkit for building and manipulating hidden Markov models and is widely used for speech recognition research. HTK can be obtained through

  URL:`http://htk.eng.cam.ac.uk`

- *sclite* scoring software

  This software comes with the NIST Scoring Toolkit (SCTK) that can be downloaded at `http://www.nist.gov/speech/tools/`.

- SRI language modeling toolkit

  SRILM is a toolkit for building and applying statistical language models (LMs), primarily for use in speech recognition, statistical tagging and segmentation. It has been under development in the SRI Speech Technology and Research Laboratory since 1995. See also Stolcke (2002).

  URL: `http://www.speech.sri.com/projects/srilm/`

- NIST tools

  Including evaluation tools, language technology tools (e.g., audio segmentation), and corpus building tools.

  URL: `http://www.nist.gov/speech/tools/`

Appendix D

Speech corpora

D.1 TNO-NRC corpus

The TNO-NRC speech database was created at *TNO Human Factors* in Soesterberg, The Netherlands and consists of 52 speakers (26 male, 26 female) reading lines from a Dutch newspaper (*NRC Handelsblad*). All speech files had been recorded using a Sennheiser HMD 414-6 close-talking microphone. The corpus contains almost 7 hours of speech.

D.2 TNO-BN corpus

Also at *TNO Human Factors* in Soesterberg, The Netherlands, the TNO-BN speech database was created. Some 20 hours of Dutch broadcast news shows (*NOS Acht uur journaal*) were manually segmented and transcribed at the word level. A first version of the TNO-BN corpus, also referred to in this research, contained 14 hours of speech.

D.3 Groningen corpus

The Groningen corpus is a corpus of read speech of 238 speakers who read 2 short texts, 23 short sentences containing all possible vowels and all possible consonants and consonant clusters in Dutch, 20 numbers and 16 monosyllabic words containing all possible vowels in Dutch. The recordings were made in a silent room with living room acoustics using a B&K 4003 microphone. Files were down-sampled from 48 kHz to 16 kHz. The corpus contains over 20 hours of speech. The speech data was gathered by Drs. A. M. Sulter, as part of an NWO project.

D.4 Speech Styles corpus

The Speech Styles database was made by SPEX and contains spontaneous speech (one monologue per speaker), semi-spontaneous speech (five picture descriptions per speaker) and read speech (five per speaker) of 130 speakers in total.

D.5 Spoken Dutch Corpus (CGN)

Release 5 of the Spoken Dutch Corpus (Oostdijk, 2000) was published in April 2002 and contains more than 450 hours of orthographically transcribed speech of a target of 1000 hours of speech. A part of the data is enriched with part-of-speech tags.

Bibliography

Abberley, D., Renals, S., Cook, G., and Robinson, T. (1999). Retrieval of Broadcast News Documents with the THISL System. In *Proceedings TREC-7*, pages 181–190.

Adda, G., Jardino, M., and Gauvain, J. (1999). Language Modelling for Broadcast News Transcription. In *Eurospeech'99*, pages 1759–1762, Budapest.

Adda-Decker, M. and Adda, G. (2000). Morphological decomposition for ASR in German. In *Workshop on Phonetics and Phonology in ASR*, Saarbrücken, Germany.

Adda-Decker, M. and Lamel, L. (2000). The Use of Lexica in Automatic Speech Recognition. In Eynde, F. v. and Gibbon, D., editors, *Lexicon Development for Speech and Language Processing*. Kluwer Academic.

Atal, B. S. and Hanauer, S. (1971). Speech analysis and synthesis by prediction of the speech wave. *Journal of the Acoustical Society of America*, 50:637–655.

Auzanne, C., Garofolo, J., Fiscus, J., and Fisher, W. (2000). Automatic language model adaptation for spoken document retrieval. In *Proceedings of RIAO 2000, Content-Based Multimedia Information Access*, pages 132–141.

Baan, J., van Ballegooij, A., Geusenbroek, J. M., den Hartog, J., Hiemstra, D., List, J., Patras, I., Raaijmakers, S., Snoek, C., Todoran, L., Vendrig, J., de Vries, A., Westerveld, T., and Worring, M. (2002). Lazy Users and Automatic Video Retrieval Tools in (the) Lowlands. In *Proceedings of the tenth Text Retrieval Conference (TREC-2001)*. NIST Special Publication.

Baayen, R., Piepenbrock, R., and van Rijn, H. (1993). The celex lexical database on cd-rom.

Bahl, L. R., Brown, P. F., de Souza, P. V., and Mercer, R. L. (1989). A tree-based statistical language model for natural language speech recognition. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 37, pages 1001–1008.

Baum, L. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8.

Bellegarda, J. R. (2000). Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1):76–84.

Berton, A., Fetter, P., and Regel-Brietzmann, P. (1996). Compound words in large-vocabulary german speech recognition systems. In *Proc. ICSLP '96*, volume 2, pages 1165–1168, Philadelphia, PA.

Bosch, A. P. J. van den (1997). Learning to pronounce written words, A study in inductive language learning. Master's thesis, University of Maastricht, The Netherlands.

Bosch, A. van den, and Daelemans, W. (1993). Dataoriented methods for grapheme-to-phoneme conversion. In *Proceedings of European Chapter of ACL*, pages 45–53, Utrecht.

Bouma, G. (2000). A finite-state and data-oriented method for grapheme to phoneme conversion. In *Proceedings of the first conference of the North-American Chapter of the Association for Computational Linguistics*, pages 303–310, Somerset, NJ. Association for Computational Linguistics.

Bouma, G. and Schuurman, I. (1998). De positie van het Nederlands in Taal- en Spraaktechnologie. Technical report, Rijksuniversiteit Groningen and Katholike Universiteit Leuven. `http://www.taalunie.org/_/publicaties/rapporten/01/rapport.ps.gz`.

Bourland, H. and Morgan, N. (1994). *Connectionist Speech Recognition-A Hybrid Approach*. Kluwer Academic.

Brown, M. G., Foote, J., Jones, G., Jones, K. S., and Young, S. J. (1995). Automatic Content-based Retrieval of Broadcast News. In *Proceedings of the third ACM international conference on Multimedia*, pages 35–43, San Francisco. ACM Press.

Brown, M. G., Foote, J. T., Jones, G. J. F., Jones, K. S., and Young, S. J. (1996). Open-Vocabulary Speech Indexing for Voice and Video Mail Retrieval. In *Proceedings of ACM Multimedia*, pages 307–316.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., Lai, J. C., and Mercer, R. L. (1992). An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18:31–40.

Busser, G. (1998). TreeTalk-D: A Machine Learning approach to Dutch word pronunciation. In Sojka, P., Matousek, V., Pala, K., and Kopecek, I., editors, *Proceedings TSD Conference*, pages 3–8, Masaryk University, Czech Republic.

Carey, M. and Parris, E. (1995). Topic spotting with task independent models. In *Proceedings of Eurospeech '95*, pages 2133–2136, Madrid, Spain.

Carter, D., Kaja, J., Neumeyer, L., Rayner, M., Weng, F., and Wirén, M. (1996). Handling Compound Nouns in a Swedish Speech-Understanding System. In *Proceedings of ICSLP-96*, volume 1, pages 26–29, Philadelphia, PA.

Chen, S. F., Beeferman, D., and Rosenfeld, R. (1998). Evaluation metrics for language models. In *Proceedings of the DARPA broadcast News Transcription and Understanding workshop*, pages 275–280.

Chen, S. F. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, Cambridge Massachusetts.

Christie, J. (1996). Completion of tno-abbot research project. Technical report, Cambridge University.

Cieri, C., Graff, D., Libermann, M., Martey, N., and Strassel, S. (1999). The tdt-2 text and speech corpus. In *Proceedings of the DARPA Broadcast News Workshop*.

Clarkson, P. and Rosenfeld, R. (1997). Statistical language modelling using the CMU-Cambridge toolkit. In *Eurospeech-97*, pages 2707–2710.

Clarkson, P. R. and Robinson, A. J. (1997). Language model adaptation using mixtures and an exponentially decaying cache. In *Proceedings of ICASSP-97*, volume 2, pages 799–802, Munich.

Cook, G. D. and Robinson, A. J. (1997). The 1997 ABBOT system for the transcription of broadcast news. In *1997 DARPA Speech Recognition Workshop*.

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. John Wiley & Sons, New York.

Dalen-Oskam, K. van, Geirnaert, D., and Kruyt, J. (2002). Text Typology and Selection Criteria for a Balanced Corpus: the Integrated Language Database of 8th-21st-Century Dutch. In Braasch, A. and Povlsen, C., editors, *Proceedings of the Tenth EURALEX International Congress, EURALEX 2002*, volume I, pages 401–406, Copenhagen, Denmark.

Eakins, J. P. and Graham, M. E. (1999). Automatic Image Content Retrieval. Technical report, Institute for Image Data Research, University of Northumbria, Newcastle. report to the JISC Technology Applications Programme (URL: `http://www.unn.ac.uk/iidr/report.html`).

Ekkelenkamp, R., Kraaij, W., and van Leeuwen, D. (1999). TNO TREC7 site reports: SDR and filtering. In *Proceedings of TREC-7*, pages 519–526.

Foote, J. T., Jones, G. J. F., Jones, K. S., and Young, S. J. (1995). Talker-independent keyword spotting for information retrieval. In *Eurospeech 1995*, volume 3, pages 2145–2148, Madrid.

Fosler-Lussier, E. (1999). *Dynamic Pronunciation Models for Automatic Speech Recognition*. PhD thesis, University of California, Berkeley, CA, USA.

Garofolo, J., Auzanne, C., and Voorhees, E. (2000). The TREC SDR Track: A Success Story. In *Eighth Text Retrieval Conference*, pages 107–129, Washington.

Garofolo, J., Lard, J., and Voorhees, E. (2001). TREC-9 Spoken Document Retrieval Track. Conference Slides.

Gauvain, J., Adda, G., Lamel, L., and Adda-Decker, M. (1997). Transcribing Broadcast News: The LIMSI Nov96 Hub4 System. In *Proc. ARPA Speech Recognition Workshop, Chantilly, Virginia*, pages 56–63.

Gauvain, J., Lamel, L., and Adda-Decker, M. (1995). Developments in Continuous Speech Dictation using the ARPA WSJ Task. In *IEEE-ICASSP*, pages 65–68, Detroit.

Gauvain, J.-L., Lamel, L., Barras, C., Adda, G., and de Kercadio, Y. (2000). The LIMSI SDR system for TREC-9. In *TREC-9*, Washington.

Goldman, J., Renals, S., Bird, S., de Jong, F., Stewart, C., Frederico, M., Fleischhauer, C., Lamel, L., Kornbluh, M., Sebastiani, F., Oard, D. W., and Wright, R. (2003). Spoken word archive group (swag) final report: Eu-us working group on spoken-word audio collections. url-http://www.dcs.shef.ac.uk/spandh/projects/swag/reports.html.

Good, I. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3 and 4):237–264.

Goodrum, A. and Rasmussen, E. (2000). Sound and speech in information retrieval: an introduction. Bulletin of the American Society for Information Science. (URL: `http://www.asis.org/Bulletin/June-00/godrumrasmussen.html`).

Gotoh, Y. and Renals, S. (2000). Topic-based mixture language modelling. *Natural Language Engineering*, pages 5:355–375.

Graff, D. (2002). An overview of Broadcast News Corpora. *Speech Communications*, 37:15–26.

Harman, D. and Voorhees, E., editors (1997). *The Fifth Text REtrieval Conference (TREC-5)*, Gaithersburg. NIST.

Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustic Society of America*, 87(2-3):1738–1752.

Heuvel, H. van den,Kuijk, D. van, and Boves, L. (2003). Modeling lexical stress in continuous speech recognition for dutch. *Speech Communication*, 40:335–350.

Heuven, V. van, and Pols, L., editors (1993). *Analysis and synthesis of speech; strategic research towards high-quality text-to-speech generation*, chapter MORPHON, lexiocn-based text-to-phoneme conversion and phonological rules. Berlin: Mouton de Gruyter.

Hiemstra, D. (2001). *Using language models for information retrieval*. PhD thesis, University of Twente.

Hochberg, M., Cook, G., Renals, S., and Robinson, T. (1994). Connectionist model combination for large vocabulary speech recognition. *IEEE Proc. Neural Networks for Signal Processing*, 4:269–278.

Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32:67–72.

James, D. (1995). *The Application of Classical Information Retrieval Techniques to Spoken Documents*. PhD thesis, Downing College, UK.

James, D. A. and Young, S. J. (1994). A fast lattice-based approach to vocabulary independent wordspotting. In *Proceedings of ICASSP '94*, pages 377–380, Adelaide, Australia.

Jeffreys, H. (1948). *Theory of Probability*. Clarendon Press, Oxford, second edition edition.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. In *Proceedings of the IEEE*, volume 64:4, pages 532–557.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts.

Jelinek, F., Mercer, R., and Bahl, L. (1975). Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, IT-21(3):250–256.

Jelinek, F. and Mercer, R. L. (1980). Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands.

Johnson, S., Jourlin, P., Moore, G., Jones, K. S., and Woodland, P. (1998). Spoken Document Retrieval for TREC-7. In *TREC-7*, Washington.

Johnson, S., Jourlin, P., Spärck Jones, K., and Woodland, P. (1999). Spoken Document Retrieval for TREC-8 at Cambridge University. In *TREC-8*, pages 197–206, Washington.

Johnson, S., Jourlin, P., Spärck Jones, K., and Woodland, P. (2000). Spoken Document Retrieval for TREC-9 at Cambridge University. In *TREC-9*.

Jones, G. J. F., Foote, J. T., Jones, K. S., and Young, S. J. (1996). Retrieving Spoken Documents by Combining Multiple Index Sources. In *Proceedings of the 1996 ACM SIGIR Conference Research and Development in Information Retrieval*, pages 30–38, Zurich, Switzerland.

Jong, F. de, Gauvain, J., Hartog, J. den, and Netter, K. (1999). Olive: Speech based video retrieval. In *CBMI'99*, Toulouse.

Jong, F. de, Gauvain, J., Hiemstra, D., and Netter, K. (2000). Language-Based Multimedia Information Retrieval. In *6th RIAO Conference*, Paris.

Jost, U., Heine, H., and Evermann, G. (1997). What is wrong with the Lexicon – An attempt to Model Pronunciation Probabilistically. In *Eurospeech '97*, pages 2475–2478.

Jourlin, P., Johnson, S., Jones, K. S., and Woodland, P. (1999). General query expansion techniques for spoken document retrieval. In *Proc. ESCA Workshop on Extracting Information from Spoken Audio*, pages 8–13, Cambridge, UK.

Jurafsky, D. and Martin, J. H. (2000). *An introduction to Speech and language Processing,Computational Linguistics, and Speech Recognition*. Prentice Hall.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transcactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400–401.

Kershaw, D., Hochberg, M., and Robinson, A. (1996). Context-Dependent Classes in a Hybrid Recurrent Network-HMM Speech Recognition System. In *Advances in Neural Information Processing Systems*, volume 8. MIT press.

Kessens, J. (2002). *Making a difference: On automatic transcription and modeling of Dutch pronunciation variation for automatic speech recognition*. PhD thesis, Katholieke Universiteit Nijmegen.

Kienappel, A. K. and Kneser, R. (2001). Designing very compact decision trees for grapheme-to-phoneme transcription. In *Proceedings Eurospeech 2001 (Scandinavia)*, pages 1911–1914.

Klakow, D. and Peters, J. (2002). Testing the correlation of word error rate and perplexity. *Speech Communications*, 38:19–28.

Kneser, R. and Ney, H. (1995). Improved back-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181–184.

Kraaij, W., Gent, J. van, Ekkelenkamp, R., and Leeuwen, D. van (1998). Phoneme based spoken document retrieval. In *Proceedings of the fourteenth Twente Workshop on Language Technology TWLT-14*, pages 141–153, University of Twente.

Kubala, F., Schwartz, R., Stone, R., and Weischedel, R. (1998). Named entity extraction form speech. In *Proceedings of DARPA Broadcast News Transcription And Understanding Workshop*.

Kuhn, R. and Mori, R. D. (1990). A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583.

Lamel, L., Gauvain, J., and Adda, G. (2001). Investigating lightly supervised acoustic model training. In *ICASSP-01*, Salt Lake City.

Larson, M., Willett, D., Köhler, J., and Rigoll, G. (2000). Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parlianmentary speeches. In *6th Int. Conference on Spoken Language Processing (ICSLP)*, Beijing, China.

Leeuwen, D. A. van, Kraaij, W., and Ekkelenkamp, R. (1999). Prediction of keyword spotting performance based on phonemic contents. In Robinson, T. and Renals, S., editors, *Proceedings of the ESCA ETRW workshop Accessing Information in Spoken Audio*, pages 73–77.

List, J., van Ballegooij, A., and Vries, A. de (2001). Known-Item Retrieval on Broadcast TV. Technical Report INS-R0104, Research Institute for Mathematics and Computer Science (CWI).

Mana, F., Massimino, P., and Pacchiotti, A. (2001). Using machine learning techniques for grapheme to phoneme transcription. In *Proceedings Eurospeech 2001 (Scandinavia)*, pages 1915–1918.

Maybury, M., Merlino, A., and Rayson, J. (1997). Segmentation, Content Extraction and Visualization of Broadcast News Video using Multistream Analysis. In *AAAI Spring Symp.*, Stanford.

Monz, C. and Rijke, M. de (2002). Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In *Proceedings CLEF 2001*. Springer.

Ney, H., Essen, U., and Kneser, R. (1994). On structuring probabilistic dependences in stochastic language modeling. *Computer, Speech and Language*, 8:1–38.

Ng, K. (2000). *Subword-based Approaches for Spoken Document Retrieval*. PhD thesis, Massachusetts Institute of Technology.

Nguyen, L., Matsoukas, S., Davenport, J., Kubala, F., Schwartz, R., and Makhoul, J. (2002). Progress in transcription of Broadcast News using Byblos. *Speech Communications*, 38:213–230.

Oostdijk, N. (2000). The Spoken Dutch Corpus. Overview and first evaluation. In Gravilidou, M., Carayannis, G., Markantonatou, S., Piperidis, S., and Stainhaouer, G., editors, *Second International Conference on Language Resources and Evaluation*, volume II, pages 887–894.

Ordelman, R., Hessen, A. van, and Leeuwen, D. van (1999a). Improving Recognition Performance Using Co-articulation Rules on the Phrase Level: A First Approach. In *Proceedings of the 14th International Congress of Phonetic Sciences, San Francisco*, pages 1641–1644.

Ordelman, R., Hessen, A. van, and Leeuwen, D. van (April 1999b). Dealing with phrase Lecel Co-raticulation (PLC) in Speech Recognition: A First Approach. In *Proceedings ESCA ETRW Workshop Accessing Information in Spoken Audio*, pages 64–68.

Ordelman, R. (2000). Zoeken in historisch videomateriaal: ECHO project. Informatie Professional. ISSN 1385-5328, jrg. 4, nummer 12, 24-29.

Ordelman, R., Melis, P., and Hessen, A. van (2001a). The Van Dale Grafheme-to-phoneme Converter. Technical report, University of Twente, Parlevink Group.

Ordelman, R., Hessen, A. van, and Jong, F. de (2001b). Lexicon Optimization for Dutch Speech Recognition in Spoken Document Retrieval. In *Proceedings of Eurospeech 2001*, pages 1085–1088, Aalborg.

Ordelman, R., Hessen, A. van, and Jong, F. de (2001c). Speech Recognition Issues for Dutch Spoken Document Retrieval. In *Proceedings of 4th International Conference TSD 2001*, pages 258–265.

Ordelman, R., Hessen, A. van, Jong, F. de, and Leeuwen, D. van (2001d). Speech Recognition for Dutch Spoken Document Retrieval. In *Proceedings of CBMI 2001*, Brescia.

Owen, C. and Makedon, F. (1999). Cross-modal information retrieval. In Furht, B., editor, *Handbook of Multimedia Computing*, pages 403–423. CRC Press, Boca Raton.

Pallett, D. S. (2002). The role of the National Institute of Standards and Technology in DARPA's Broadcast News continuous speech recognition research program. *Speech Communication*, 37:3–14.

Paul, D. and Baker, J. (1992). The design for the Wall Street Journal-based CSR corpus. In *Proc. Fifth DARPA Speech and Natural Language Workshop*, pages 357–362. Morgan Kaufmann Publishers, Inc.

Petković, M. (2003). *Content-based video retrieval supported by database technology*. PhD thesis, University of Twente.

Pfeiffer, S. (1999). *Information Retrieval aus digitalisierten Audiospuren von Filmen*. PhD thesis, Universität Mannheim.

Pohlmann, R. and Kraaij, W. (1996). Improving the precision of a text retrieval system with compound analysis. In Landsbergen, J., Odijk, J., van Deemter, K., and van Zanten, G. V., editors, *Proceedings of the 7th Computational Linguistics in the Netherlands (CLIN 1996)*, pages 115–129.

Rabiner, L. and Juang, B. (1993). *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ.

Renals, S. (1996). Phone deactivation pruning in large vocabulary continuous speech recognition. In *IEEE Signal Processing Letters 3*, pages 4–6.

Renals, S. and Hochberg, M. (1999). Start-synchronous search for large vocabulary continuous speech recognition. In *IEEE Transactions on Speech and Audio Processing*, volume 7, pages 542–553.

Rijsbergen, C. (1979). *Information Retrieval*. London Butterworths.

Robertson, S. and Spärck-Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146.

Robertson, S., Walker, S., and Beaulieu, M. (1998). Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track. In *Proceedings of TREC-7*.

Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In Croft, W. B. and van Rijsbergen, C. J., editors, *Proceedings of the 17th International Conference on Research and Development in Information Retrieval*, pages 232–241, Dublin. Springer–Verlag.

Robinson, A., Cook, G., Ellis, D., Fosler-Lussier, E., Renals, S., and Williams, D. (2002). Connectionist Speech Recognition of Broadcast News. *Speech Communication*, 37:27–45.

Robinson, A. J. (1994). An Application of Recurrent Nets to Phone Probability Estimation. In *IEEE Transactions on Neural Networks, vol. 5*, pages 298–305.

Robinson, T., Abberley, D., Kirby, D., and Renals, S. (1999). Recognition, Indexing and Retrieval of British Broadcast News with the THISL System. In *Proccedings of Eurospeech '99*, volume 3, pages 1267–1270.

Robinson, T. and Christie, J. (1998). Time-first search for large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 829–832.

Robinson, T., Hochberg, M., and Renals, S. (1996). *The use of recurrent networks in continuous speech recognition*, chapter 7, pages 233–258. Kluwer Academic Publishers.

Ordelman, and Jong, F. de (2003). Compound decomposition in Dutch large vocabulary speech recognition. In *Proceedings of Eurospeech 2003*, Geève, Switzerland.

Rose, R. C., Chang, E. I., and Lippmann, R. (1991). Techniques for information retrieval from voice messages. In *Proceedings ICASSP '91*, pages 317–320, Toronto, Canada.

Rosenfeld, R. (1995). Optimizing Lexical and N-gram Coverage Via Judicious Use of Linguistic Data. In *Eurospeech-95*, pages 1763–1766.

Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval.* New York: McGraw-Hill.

Sankar, A., Gadde, V. R. R., Stolcke, A., and Weng, F. (2002). Improved modeling and efficiency for automatic transcription of Broadcast News. *Speech Communication*, 37:133–158.

Schuurman, L. (1997). Anno: a multi-functional flemish text corpus. In et al., J. L., editor, *Papers from the Seventh CLIN meeting*, pages 161–176, IPO, Technische Universiteit Eindhoven.

Schwartz, R. and Chow, Y.-L. (1990). The n-best algorithm: An efficient and exact procedure for finding the n most likely sentence hypotheses. In *IEEE ICASSP-90*, volume 1, pages 81–84.

Sekine, S. and Grisham, R. (1995). NYU language modeling experiments for the 1995 CSR evaluation. In *Proc. ARPA Spoken Language Sys. Technology Workshop*, pages 123–128, Harriman, NY.

Seymore, K., Chen, S., Eskenazi, M., and Rosenfeld, R. (1997). "language and pronunciation modeling in the cmu 1996 hub 4 evaluation". In *"Proceedings of DARPA Speech Recognition Workshop"*, pages 141–146, Chantilly, Virginia.

Seymore, K. and Rosenfeld, R. (1997). Using story topics for language model adaptation. In *Proceedings of Eurospeech.*

Sheridan, P. and Ballerini, J. P. (1996). Experiments in multilingual information retrieval using the SPIDER system. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 58–65, Zurich, Switzerland.

Siivola, V., Kurimo, M., and Lagus, K. (2001). Large vocabulary statistical language modeling for continuous speech recognition in finnish. In *Proceedings of the 7th European Conference on Speech Communication and Technology*, volume 1, pages 737–730.

Singhal, A. and Pereira, F. C. N. (1999a). Document expansion for speech retrieval. In *Research and Development in Information Retrieval*, pages 34–41.

Singhal, A. and Pereira, F. C. N. (1999b). Document Expansion for Speech Retrieval. In *Proceedings SIGIR '99*, pages 34–41.

Smeaton, A. F., Morony, M., Quinn, G., and Scaife, R. (1998). Taiscéalaí: Information Retrieval from an Archive of Spoken Radio News. In *Proceeding of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL2)*, pages 429–442, Crete.

Spink, A. and Saracevic, T. (1997). Interaction in information retrieval: selection and effectiveness of search terms. *Journal of the American Society Information Science*, 48(8):741–761.

Stoianov, I. and Nerbonne, J. (1999). Connectionist Grapheme to Phoneme Conversion: Exploring Distributed Representations. In *CLIN '99: Computational Linguistics in the Netherlands*, Utrecht.

Stolcke, A. (1998). Entropy-based pruning of backoff language models. In *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, pages 270–274, Lansdowne, VA. Morgan Kaufmann.

Stolcke, A. (2002). SRILM – An Extensible Language Modeling Toolkit. In *International Conference Spoken Language Processing*, Denver, Colorado. `http://www.speech.sri.com/projects/srilm/`.

Strik, H. and Cucchiarini, C. (1999). Modeling pronunciation variation for ASR: A survey of the literature. *Speech Communication*, 29:225–246.

Velthausz, D. D. (1998). *Cost-effective network-based multimedia information retrieval*. PhD thesis, University of Twente.

Veth, J. de (2001). *On Speech Sound model Accuracy*. PhD thesis, University of Nijmegen, The Netherlands.

Voorhees, E. (2000). The TREC-8 question answering track report. In *Proceedings of the eight Text REtrieval Conference (TREC-8)*, pages 77-82. NIST Special Publication 500-246.

Voorhees, E., Garofolo, J., and Jones, K. S. (1997). The TREC-6 Spoken Document Retrieval Track. In *Proceedings DARPA Speech Recognition Workshop*.

Vosse, T. G. (1994). *The Word Connection*. PhD thesis, University of Leiden, The Netherlands. Neslia Paniculata Uitgeverij, Enschede.

Vries, A. P. de (1999). *Content and Multimedia Database Management Systems*. PhD thesis, Center for Telematics and Information Technology, University of Twente.

Wayne, C. (2000). Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation. In *Language Resources and Evaluation Conference (LREC)*, pages 1487–1494.

Weintraub, M., Aksu, Y., Dharanipragada, S., Khudanpur, S., Ney, H., Prange, J., Stolcke, A., Jelinek, F., and Shriberg, E. (1996). Lm95 project report: Fast training and portability. Research Note 1, Center for Language and Speech Processing, Johns Hopkins University, Baltimore.

Werbos, P. (1990). Backpropagation through time: what it does and how to do it. In *IEEE*, volume 78, pages 1150–1160.

Wester, M. (2002). *Pronunciation Variation Modeling for Dutch Automatic Speech Recognition*. PhD thesis, Katholieke Universiteit Nijmegen.

Westerveld, T. (2002). Probabilistic Multimedia Retrieval. In *Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002)*.

Witbrock, M. J. and Hauptmann, A. G. (1998). Speech recognition for a digital video library. *Journal of the American Society of Information Science*, 49(7):619–632.

Witten, I. and Bell, T. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094.

Woodland, P. (2002). The development of the HTK Broadcast News transcription system: an overview. *Speech Communication*, 37:47–67.

Woodland, P., Johnson, S., Jourlin, P., and Jones, K. S. (2000). Effects of Out of Vocabulary Words in Spoken Document Retrieval. In *2000 ACM SIGIR Conference*, pages pp 372–374, Athens, Greece.

Woordenlijst Nederlandse Taal, I. (1995). *Samengesteld door het Instituut voor Nederlandse Lexicografie in opdracht van de Nederlandse Taalunie*. SDU Uitgevers, Den Haag.

Summary

As data storage capacities grow to nearly unlimited sizes thanks to ever ongoing hardware and software improvements, an increasing amount of information is being stored in multimedia and spoken-word collections. Assuming that the intention of data storage is to use (portions of) it some later time, these collections must also be searchable in one way or another. For multimedia and spoken-word collections, traditional text-oriented information retrieval (IR) strategies inevitably fall short, as the amount of textual information included with these types of documents is usually very limited. However, when automatic speech recognition (ASR) can be used to convert the speech occurring in these documents into text, textual representations can be created that in turn can be searched using the traditional text-based search strategies. As ASR systems label recognized words with exact time information as a standard accessory, detailed searching within multimedia and spoken-word collections can be enabled. This type of retrieval is commonly referred to as Spoken Document Retrieval (SDR).

Typically, large vocabulary speaker independent continuous speech recognition systems (LVCSR) are deployed for creating textual representations of the spoken audio in multimedia an spoken-word collections. For Dutch however, such a system was not available when this research was started. As creating a Dutch system from scratch was not feasible given the available resources, an existing English system, refered to as the *ABBOT* system, was ported to Dutch. A significant part of this thesis is dedicated to a complete run-down of the porting work, involving the collection and preparation of suitable training data and the actual training and evaluation of the acoustic models and language models. The broadcast news domain was chosen as domain of focus, as this domain has also been extensively used as a benchmark domain for both international ASR research and SDR. A complicating factor for ASR in the news domain, is that word usage is highly variable. As a consequence, besides using large vocabularies, it is important to adjust these vocabularies regularly, so that they reflect the content of the news programs well. Therefore, it has been investigated which word selection strategies are best suited for making these vocabulary adjustments. Moreover, as dynamic vocabularies require a flexible generation of accurate word pronunciations, the development of a grapheme-to-phoneme converter is addressed. Another vocabulary related issue that

is investigated, stems from a well-known characteristic of the Dutch language, word compounding: Dutch words can almost freely be joined together to form new words. As a result of this phenomenon, the number of distinct words in Dutch is relatively large, which reduces the coverage of vocabularies compared to those of the same size of other languages, such as English, that do not have word compounding. This thesis investigates whether splitting Dutch compound words could be a remedy for the relatively limited coverage of vocabularies, so that ASR performance could be improved.

Next to a brief history of SDR research and a review of possible SDR approaches, this thesis demonstrates the use of a Dutch LVCSR in SDR by providing an illustrative example of an SDR evaluation given a collection of Dutch broadcast news shows. It is shown that Dutch speech recognition can successfully be deployed for content-based retrieval of broadcast news programs. The experience obtained with the research described in this thesis, and the experience that will emerge from future research efforts must contribute to the long-term accessibility of the increasing amount of information being stored in Dutch multimedia and spoken-word collections.

Samenvatting (in Dutch)

Met de snelle vooruitgang op het gebied van computersoftware en -hardware blijft de beschikbare opslagcapaciteit maar groeien. Daarom wordt steeds meer informatie wordt opgeslagen in multimedia- en audiocollecties, in plaats van enkel en alleen in tekstbestanden. Om (delen van) eenmaal opgeslagen informatie later opnieuw te kunnen gebruiken of te kunnen raadplegen, is het van belang om op de één of andere manier in deze collecties te kunnen zoeken. Standaard, op tekst gebaseerde zoekmethoden zijn in principe niet geschikt voor het zoeken in dit soort collecties, omdat de hoeveelheid tekstuele informatie, indien beschikbaar, meestal maar beperkt is. Maar, wanneer voor dit soort collecties spraakherkenning kan worden ingezet om spraak om te zetten naar tekst, kunnen de tekstuele representaties vervolgens doorzocht worden met behulp van de standaard zoektechnieken. Omdat spraakherkenningssystemen voor elk herkend woord exact het tijdstip kunnen aangeven waarop het woord in de audio voorkomt, wordt het mogelijk om gericht te zoeken naar spraakfragmenten. Deze manier van zoeken kan worden aangeduid als spraakgerichte retrieval, in het Engels "*spoken document retrieval* (SDR)."

Voor het maken van tekstuele representaties van multimedia- en audiodocumenten maken spraakherkenners doorgaans gebruik van een groot woordenboek, geschikt voor lopende spraak en onafhankelijk van een enkele spreker (in het Engels aangeduid als *large vocabulary speaker independent continuous speech recognition (LVCSR) systems*). Voor het Nederlands bestond er aan het begin van dit promotieonderzoek echter nog niet zo'n spraakherkenningssysteem. Omdat het van de grond af opbouwen van zo'n systeem niet doenlijk was gegeven de beschikbare tijd, werd een bestaande Engels spraakherkenner, *ABBOT*, omgezet naar een versie voor het Nederlands. Een belangrijk deel van dit proefschrift is gewijd aan deze omzetting, die ondermeer bestond uit het vergaren en prepareren van geschikt trainingsmateriaal, en het trainen en evalueren van de akoestische modellen en de taalmodellen. Omdat radio- en televisienieuws wereldwijd wordt gebruikt in evaluaties van spraakherkenningsonderzoek en onderzoek naar spraakgerichte retrieval, is ook voor dit proefschrift het nieuwsdomein gekozen als testdomein. Een complicerende factor voor spraakherkenning in het nieuwsdomein is echter dat het woordgebruik er nogal variabel is. Daardoor zijn er niet alleen grote woordenboeken nodig, maar moeten deze ook regelmatig worden aangepast aan de inhoud van de nieuwsuitzendingen. Dit proefschrift beschrijft daarom welke woordselectieprocedures het meest geschikt zijn om de woordenboeken te kunnen aanpassen aan de periode waarop het nieuws betrekking heeft. Omdat ook uitspraakinformatie hierbij beschikbaar moet zijn, is ook een grafeem-naar-foneem omzetter ontwikkeld. Een ander onderzoeksthema dat wordt behandeld hangt samen met het feit dat het Nederlands samenstellingen kent: woorden kunnen bijna ongelimiteerd aan elkaar worden geplakt om zodoende nieuwe woorden te creëren. Hierdoor is het aantal verschillende woorden in het Nederlands relatief groot, waardoor de dekking van woordenboeken minder is in vergelijking tot talen zonder samenstellingen, zoals het Engels. Dit proefschrift behandelt de vraag of het splitsen van samenstellingen een oplossing kan zijn om dekking van de woordenboeken te vergroten, zodat de prestaties van de spraakherkenner kunnen worden verbeterd.

Naast een korte geschiedenis van onderzoek naar spraakgerichte retrieval en een overzicht van mogelijke benaderingen op dit gebied, behandelt dit proefschrift de ontwikkeling en toepassing van Nederlandse spraakherkenning in spraakgerichte retrieval. Een illustratief voorbeeld laat zien dat Nederlandse spraakherkenning succesvol kan worden ingezet voor het zoeken in een collectie van televisiejournaals. De ervaringen en de resultaten die uit dit onderzoek en toekomstig onderzoek volgen, moeten een belangrijke bijdrage leveren aan de toekomstige toegankelijkheid van de steeds maar groeiende hoeveelheid aan informatie die wordt opgeslagen in Nederlandse multimedia en audiocollecties.

Curriculum Vitae

Roeland Ordelman was born on March 20th, 1969 in Oud-Beijerland, the Netherlands. In 1989 he graduated from the CSG ”Willem van Oranje” and started his quest for a suitable trade to enable him to live happily ever after. Studying Law at the University of Utrecht did not appear to be the right choice so he prepared himself for a music study at the Conservatory in The Hague. Although, fortunately, some talent was acknowledged there, he soon realised that he would not end up among the world’s most famous in music. Unwilling to devote his life to a very uncertain musical career, his passion for languages and literature made him decide to study Spanish language and literature at Utrecht University. After some muddling around, he discovered what science could be all about, thanks to an inspiring lecture on linguistics by Jan Don, and switched to the study of Phonetics. In 1998 he completed his Master’s thesis on a subject in the field of psycholinguistics, “Vowels and Consonants in Word Reconstruction” and passed his *Doctoraal Examen* (equivalent to obtaining an MA degree), having specialised in Automatic Speech Recognition. From July 1998 to July 2003, he was employed as a research assistant at the Language, Knowledge and Interaction group (TKI) of the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente. He worked on three R&D projects aiming at the development of automatic speech recognition for Dutch as a tool for the retrieval of multimedia content: DRUID, ECHO and Waterland. This thesis reflects the work carried out within these projects. From October 2003, he will continue his work as researcher at the University of Twente.

SIKS DISSERTATION SERIES

1998-1 .... Johan van den Akker (CWI), *DEGAS - An Active, Tempo-//[.3cm] ral Database of Autonomous Objects*

1998-2 .... Floris Wiesman (UM), *Information Retrieval by Graphi-//[.3cm] cally Browsing Meta-Information*

1998-3 .... Ans Steuten (TUD), *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*

1998-4 .... Dennis Breuker (UM), *Memory versus Search in Games*

1998-5 .... E.W.Oskamp (RUL), *Computerondersteuning bij Straftoemeting*

1999-1 .... Mark Sloof (VU), *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*

1999-2 .... Rob Potharst (EUR), *Classification using decision trees and neural nets*

1999-3 .... Don Beal (UM), *The Nature of Minimax Search*

1999-4 .... Jacques Penders (UM), *The practical Art of Moving Physical Objects*

1999-5 .... Aldo de Moor (KUB), *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*

1999-6 .... Niek J.E. Wijngaards (VU), *Re-design of compositional systems*

1999-7 .... David Spelt (UT), *Verification support for object database design*

1999-8 .... Jacques H.J. Lenting (UM), *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.*

2000-1 .... Frank Niessink (VU), *Perspectives on Improving Software Maintenance*

2000-2 .... Koen Holtman (TUE), *Prototyping of CMS Storage Management*

2000-3 .... Carolien M.T. Metselaar (UVA), *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.*

2000-4 .... Geert de Haan (VU), *ETAG, A Formal Model of Competence Knowledge for User Interface Design*

2000-5 .... Ruud van der Pol (UM), *Knowledge-based Query Formulation in Information Retrieval.*

2000-6 .... Rogier van Eijk (UU), *Programming Languages for Agent Communication*

2000-7 .... Niels Peek (UU), *Decision-theoretic Planning of Clinical Patient Management*

2000-8 .... Veerle Coupé (EUR), *Sensitivity Analyis of Decision-Theoretic Networks*

2000-9 .... Florian Waas (CWI), *Principles of Probabilistic Query Optimization*

2000-10 .... Niels Nes (CWI), *Image Database Management System Design Considerations, Algorithms and Architecture*

2000-11 .... Jonas Karlsson (CWI), *Scalable Distributed Data Structures for Database Management*

2001-1 .... Silja Renooij (UU), *Qualitative Approaches to Quantifying Probabilistic Networks*

2001-2 .... Koen Hindriks (UU), *Agent Programming Languages: Programming with Mental Models*

2001-3 .... Maarten van Someren (UvA), *Learning as problem solving*

2001-4 .... Evgueni Smirnov (UM), *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*

2001-5 .... Jacco van Ossenbruggen (VU), *Processing Structured Hypermedia: A Matter of Style*

2001-6 .... Martijn van Welie (VU), *Task-based User Interface Design*

2001-7 .... Bastiaan Schonhage (VU), *Diva: Architectural Perspectives on Information Visualization*

2001-8 .... Pascal van Eck (VU), *A Compositional Semantic Structure for Multi-Agent Systems Dynamics.*

2001-9 .... Pieter Jan 't Hoen (RUL), *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*

2001-10 .... Maarten Sierhuis (UvA), *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*

2001-11 .... Tom M. van Engers (VUA), *Knowledge Management: The Role of Mental Models in Business Systems Design*

2002-01 .... Nico Lassing (VU), *Architecture-Level Modifiability Analysis*

2002-02 .... Roelof van Zwol (UT), *Modelling and searching web-based document collections*

2002-03 .... Henk Ernst Blok (UT), *Database Optimization Aspects for Information Retrieval*

2002-04 .... Juan Roberto Castelo Valdueza (UU), *The Discrete Acyclic Digraph Markov Model in Data Mining*

2002-05 .... Radu Serban (VU), *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*

2002-06 .... Laurens Mommers (UL), *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*

2002-07 .... Peter Boncz (CWI), *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*

2002-08 .... Jaap Gordijn (VU), *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*

2002-09 .... Willem-Jan van den Heuvel(KUB), *Integrating Modern Business Applications with Objectified Legacy Systems*

2002-10 .... Brian Sheppard (UM), *Towards Perfect Play of Scrabble*

2002-11 .... Wouter C.A. Wijngaards (VU), *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

2002-12 .... Albrecht Schmidt (Uva), *Processing XML in Database Systems*

2002-13 .... Hongjing Wu (TUE), *A Reference Architecture for Adaptive Hypermedia Applications*

2002-14............Wieke de Vries (UU), *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

2002-15............Rik Eshuis (UT), *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*

2002-16............Pieter van Langen (VU), *The Anatomy of Design: Foundations, Models and Applications*

2002-17............Stefan Manegold (UVA), *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003-01............Heiner Stuckenschmidt (VU), *Ontology-Based Information Sharing in Weakly Structured Environments*

2003-02............Jan Broersen (VU), *Modal Action Logics for Reasoning About Reactive Systems*

2003-03............Martijn Schuemie (TUD), *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*

2003-04............Milan Petkovic (UT), *Content-Based Video Retrieval Supported by Database Technology*

2003-05............Jos Lehmann (UVA), *Causation in Artificial Intelligence and Law - A modelling approach*

2003-06............Boris van Schooten (UT), *Development and specification of virtual environments*

2003-07............Machiel Jansen (UvA), *Formal Explorations of Knowledge Intensive Tasks*

2003-08............Yongping Ran (UM), *Repair Based Scheduling*

2003-09............Rens Kortmann (UM), *The resolution of visually guided behaviour*

2003-10............Andreas Lincke (UvT), *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*

2003-11............Simon Keizer (UT), *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*

2003-12............Roeland Ordelman (UT), *Dutch speech recognition in multimedia information retrieval*

2003-13............Jeroen Donkers (UM), *Nosce Hostem - Searching with Opponent Models*

Index