Gefördert vom

bmb+f

Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie

Verbmobil

Verbundvorhaben

Detection and Transcription of OOV Words

Pablo Fetter

Daimler-Benz AG

Vm

Report 231

August 1998

August 1998

Pablo Fetter

Daimler-Benz AG
Research and Technology
P.O. Box 2360
89013 Ulm
Germany

Tel.: + 49 - 731 - 505 - 2136
Fax: + 49 - 731 - 505 - 4105
e-mail: `fetter@dbag.ulm.daimlerbenz.com`

Gehört zum Antragsabschnitt: 1.8 Neue Wörter

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Förderkennzeichen 01 IV 101 G gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei dem Autor.

ISSN 0949–6084

Thanks

$t < T_0$

Empecemos por lo más importante.

Mis padres siempre me hicieron creer que yo era capaz, y me enseñaron que dedicación y perseverancia siempre conducen al éxito. Criándome con amor y exigencia sentaron las bases indispensables, que un cuarto de siglo más tarde contribuyeron decisivamente a la concresión de éste trabajo.

Cuando a los once años tuve que separarme de mis padres en búsqueda de una mejor educación, mi abuela María Luisa y mi tía Cheli dejaron el Chaco y me acompañaron a Córdoba. Allá fueron ellas mis padres postizos, y crearon el entorno de amor que me permitió completar exitosamente mis estudios secundarios y universitarios.

Gudrun und Günter adoptierten mich als einen wahren Sohn und gaben mir nicht nur ein zu Hause in Hamburg sondern auch das Gefühl, daß ich zu dieser Familie gehöre. Sie halfen mir bei der Orientierung in der für mich völlig neuen Berufslandschaft in Deutschland und waren immer die ultimative Karriereberatung.

Por Katy vine a Alemania. Ella fue mi apoyo y mi guía, el pilar que me sostuvo cuando llegué a Alemania sin amigos y sin idioma. Ella me orientó y me integró a su sociedad. Ella me enseñó una buena parte del Alemán que ahora sé y tuvo la infinita persistencia, *mir stendich bei meinem zahlreiche Fählern zu corrigiern.* Cuando se me presentó la oportunidad de trabajar en Daimler-Benz, Katy me alentó a mudarme de Berlín a Ulm, y siempre me apoyó incondicionalmente soportando más de tres difíciles años de ausencia conyugal.

$t \geq T_0$

This thesis is the condensation of some of the experiences I gained in the last $3\,\frac{1}{2}$ years of work at the Department for Speech Understanding of the Daimler-Benz Research Center in Ulm, Germany. During that time I felt privileged to work

together with interesting people in the fascinating topic of speech research.

Die außerordentlich freundliche und lockere Arbeitsatmosphäre im Daimler-Benz Forschungszentrum Ulm war der inspirierende Hintergrund dieser Arbeit. Alle Kollegen in der Abteilung sowie die Vorgesetzten ermöglichten dieses.

Mein Kollege und Freund Udo Haiber hat das Bestreben nie aufgegeben, mich von der Ignoranz zu retten. Seine Integrität auf der persönlichen Seite und seine klare, strukturierte Denkweise auf der anderen Seite waren und bleiben beispielhaft für mich. Udito hat zu dieser Arbeit ständig mit Unterstützung und Motivation und häufig mit Vorschlägen und konkreten Lösungen beigetragen.

Peter Regel-Brietzmann war mehr als nur ein Betreuer für mich und begleitete mich ständig sowohl bei meiner Doktorarbeit als auch bei den ersten Schritten meines Beruflebens. Zunächst setzte sich Peter entschieden für meine Einstellung als Doktorand bei Daimler-Benz ein. Dann führte er mich in die Spracherkennung und mein Aufgabengebiet ein. Während meiner Doktorarbeit gab er mir auf der einen Seite die Möglichkeit, die Forschungsschwerpunkte vom Thema *OOV words* selber zu bestimmen. Auf der anderen Seite war er stets bereit, meine meist unreifen Ideen mit seinen breiten und zugleich tiefen Erfahrungen zu erweitern oder einzuschränken.

Diese Arbeit wäre nicht möglich gewesen ohne das Daimler-Benz Spacherkennungssystem. Alfred Kaltenmeier, Fritz Class, Peter Regel-Brietzmann und Thomas Kuhn haben die wesentlichen Teile des Systems entworfen und implementiert. André Berton hat im Rahmen seiner Diplomarbeit die im Kapitel 4 beschriebenen Ansätze mitkonzipiert und implementiert.

My room mate David Stall corrected this thesis converting the *Spangerglish* manuscript into correct English, and enriched it with valuable ideas and contributions. Unfortunately, one of David's suggestions had to be repeatedly rejected: to typeset my thesis with Fraktur typefaces... **I would have liked to make this small concession to you, David, but I believe that other readers will be thankful that I did not.** To conclude, David: Mi agradecimiento es tan profundo, que sólo lo puedo expresar en castellano. ¡Además, quería ahorrarte la correctura de al menos una oración!

Many people throughout the world had also the perseverance to read entire sections of this thesis, improving them with many interesting suggestions. Among them were: Peter Regel-Brietzmann (Daimler-Benz in Ulm, Germany), Kai Hübener (Siemens in Munich, Germany), Henrik Heine (University of Hamburg, Germany), André Berton (Daimler-Benz in Ulm, Germany), Elmar Noeth (University of Erlangen, Germany), Florian Gallwitz (University of Erlangen, Germany), Gunnar Evermann (University of Hamburg, Germany), Udo Haiber (Daimler-Benz in Ulm, Germany), Sukumar Bhattacharya (Daimler-Benz in Bangalore, India), Markus Hartmann (Daimler-Benz in Bangalore, India), and Pramod Cherukumilli (Daimler-Benz in Bangalore, India).

Professor Jürgen Schürmann unterstützte mich bei meinen Untersuchungen im Bereich der *confidence measures* und steckte mich mit seiner Begeisterung für das Thema an. Später, als ich einen zweiten Referenten für meine Dissertation gesucht habe, hat er freundlicherweise angeboten, diese Aufgabe zu übernehmen.

Professor Klaus Fellbaum ermöglichte mir erst eine Doktorarbeit an der Technischen Universität Berlin zu beginnen. Dann trug er entscheidend mit seiner Empfehlung zu meiner Einstellung als Doktorand bei Daimler-Benz bei. Schließlich half er mir mit Rat und Tat bei den zahlreichen Schwierigkeiten, mit denen ich bis zu meiner Promotionszulassung konfrontiert wurde. Professor Fellbaum gab mir darüber hinaus große Freiheiten bei der Gestaltung meiner Doktorarbeit.

Thank you very much, muchas gracias, und vielen Dank!

Abstract

This thesis deals with the problem of *Out-Of-Vocabulary words* in speech recognition. The standard response of speech recognition systems whenever they encounter such OOV words is to (silently) misrecognize them without issuing any warning to the user. In order to avoid this undesired behaviour, two different strategies are proposed.

The first strategy consists in preventing the problem, i.e. the occurrence of OOV words, and this thesis presents two ways of doing that. First, the system vocabulary is optimized using information extracted from other corpora and application domains, such that the number of expected OOV words be minimized. Using this method, the vocabulary coverage was significantly improved, especially for small vocabularies. The second method of reducing the number of OOV words consists of redefining the concept of "word" based on morphological considerations. In particular, compound words are decomposed into their constituent parts, which are used as the lexical recognition units, and then the parts are rejoined to compounds. These units reduce the vocabulary size as well as the OOV rate, and improve the recognition performance at the same time.

The second strategy consists in adapting the system to the occurrence of OOV words. This is done by incorporating explicit knowledge of OOV words into the system models. The acoustic model is complemented with additional garbage models for OOV words. The language model is especially tuned to detecting OOV words. When OOV words occur in the language-model training material, the context of OOV words can be explicitly modeled, and the system accuracy can be significantly improved by using models for OOV words. When all words in the training are included in the vocabulary, however, OOV words will not occur in the training material. In order to be able to train a language model for OOV words in this situation, we introduced a method dubbed *iterative substitution*, which simulates the occurrence of OOV words in the language-model training material. This method also produced positive experimental results.

The next step after identifying OOV words as such is to identify new words which occur more than once. For this, the recognition system has to update its models to incorporate the new words in question, and a phonetic transcription of the new words has to be found. A method dubbed *the cheap transcription algorithm* is presented, which uses information extracted directly from the word-based search to compute the phonetic transcription. Since the phonetic representation of the input is available as a by-product of the recognition, little additional computation is required for the phonetic transcription. Experiments showed that the phonetic transcriptions obtained using the cheap transcription algorithm are almost as accurate as those obtained with a phoneme recognizer at a much higher cost.

Zusammenfassung

Diese Dissertation beschäftigt sich mit dem Problem unbekannter Wörter bei der Spracherkennung. Wenn unbekannte Wörter in ein Spracherkennungssystem eingegeben werden, werden diese vom System (stillschweigend) mißerkannt. Hier werden zwei Strategien vorgeschlagen, um dieses unangenehme Systemverhalten zu vermeiden.

Die erste Strategie ist eine Vermeidungsstrategie. Zwei Methoden werden eingeführt, deren Ziel es ist, die zu erwartende OOV-Rate zu minimieren. Bei der ersten Methode werden Informationen aus fremden Korpora und Anwendungsdomänen zur Vokabularoptimierung benutzt. Diese Methode konnte die Vokabularabdeckung signifikant verbessern. Bei der zweiten Methode wird das Konzept von "Wörtern" umdefiniert und morphologische Aspekte werden mitbetrachtet. Hierbei werden Konstituenten anstelle von Ganzwörtern als lexikalische Einheit verwendet, die nach der Erkennung wieder zu Ganzwörtern zusammengefasst werden. Diese Methode führt zu einer Reduktion der OOV-Rate bei gleichzeitiger Verringerung des Vokabulars sowie zu einer Verbesserung der Erkennungsrate.

Die zweite Strategie ist eine Lösungsstrategie, wobei das System auf das Auftreten von unbekannten Wörtern getrimmt wird. Dieses wird durch Addieren von Wissen über unbekannte Wörter in die Systemmodelle gewährleistet. In das akustische Modell werden sogenannte *Müllmodelle* eingefügt. Beim Sprachmodell wird der Kontext von unbekannten Wörtern explizit modelliert. Wenn das Trainingsmaterial unbekannte Wörter enthält, kann deren Kontext beim Training direkt modelliert werden und das Systemverhalten wird dadurch signifikant verbessert. Wenn das Trainingsmaterial keine unbekannten Wörter enthält, ist eine explizite Modellierung nicht möglich. Um ein Sprachmodell für unbekannte Wörter trotzdem trainieren zu können, wurde die Methode der *iterativen Substitution* entwickelt, die das Vorkommen von unbekannten Wörtern im Trainingsmaterial simuliert. Iterative Substitution zeigte positive experimentelle Ergebnisse.

Nach der Detektion ist es wünschenswert, daß neue Wörter dem Systemvokabular hinzugefügt werden, damit sie bei wiederholtem Vorkommen wiedererkannt werden können. Dafür müssen die Systemmodelle aktualisiert werden, und eine phonetische Transkription der neuen Wörter ist notwendig. Der *Cheap Transcription Algorithm* wurde zu diesem Zweck entwickelt. Dabei werden Informationen vom normalen (Wort-) Erkennungsprozess extrahiert, um eine phonetische Transkription zu erzeugen. Da die phonetische Transkription als "Abfallprodukt" der Erkennung entsteht, wird sehr wenig zusätzlicher Rechenaufwand dafür benötigt. Dabei sind die Transkriptionen fast ebenso genau wie diejenigen eines Phonemerkenners.

Contents

List of Tables

List of Figures

1 INTRODUCTION

OOV? I don't know that word.
A friend

1.1 Out-Of-Vocabulary Words

IMAGINE you meet a person, and after you introduce yourself, the person says something like: *"drawkcab si eman ym olleh*[1]*."* As soon as you hear this utterance, you will be certain that you did not understand it. As a human being, you possess excellent speech understanding capabilities, which allow you to react appropriately. *"I am sorry, I don't understand you,"* you will probably answer.

If the same utterance is input to a computer running a state-of-the-art large-vocabulary continuous-speech dictation system, the result on your monitor will be something like: *"draw cabs imagine olden,"* which is obviously not the desired response. The task of discriminating between known and unknown speech sequences, that appears to be trivial to humans, is indeed a challenging problem in automatic speech recognition. And this is the problem I am concerned with in this thesis.

Speech recognizers are based on a vocabulary which contains all words the system knows. In isolated speech recognition (i.e. when words are spoken one at a time), the word to be recognized is basically compared with all words in the vocabulary, and the vocabulary entry that best matches the input is chosen as "the result." If the word spoken by the speaker is not in the system vocabulary, i.e. if it is an *Out-Of-Vocabulary (OOV) word*, the system will nevertheless proceed as usual, and choose the most similar entry in its vocabulary, which, of course, will be wrong.

Recognizing continuous speech is much more difficult, because the word boundaries are unknown. Therefore, virtually all words in the vocabulary have to be considered all the time, and the recognition result is chosen taking many alternatives into account, which can partially overlap in time (see Chapter 2 for a more

[1] "Hello, my name is Backward," written from right to left.

detailed explanation of this process). In this framework, detecting OOV words is extremely difficult.

Viewed from a more general perspective, the problem of OOV words is an intrinsic problem of many pattern recognition applications, a consequence of the so-called "closed-world assumption" [Schürmann, 1996]. Usually pattern recognition systems rely on a training phase, in which the characteristics of the patterns to be recognized are in some way learned and stored. During recognition, the unknown input is compared with the stored patterns, and the most similar one is chosen according to some appropriate measure of similarity. This implies that only those patterns which have been trained can be recognized, and that in the case of an unknown pattern being fed to the system, the most similar known pattern will be (mis)recognized instead. In this thesis, however, I will not pursue general-purpose approaches. I will rather concentrate on the particularities of words as patterns, and speech recognition as application.

1.2 Approaches Presented in This Thesis

Speech recognizers use various models that capture different levels of information contained in speech, including the acoustic model (for the information contained in the acoustic signal), and the language model (for the linguistic information). If these models can be modified to determine the occurrence of OOV words, OOV-word detection could be incorporated into the framework of standard speech recognizers. In Chapter 5 this approach will be studied in detail.

The nonsense example given at the beginning of this chapter does not contain any word worth remembering. In some cases, however, learning new words is indeed important for both human beings and speech recognizers. In order to learn a new word, a recognition system has to incorporate it into its models. A very important model to be enhanced is the acoustic model, and (in the framework of a recognition system based on subword units) in particular the pronunciation dictionary. In Chapter 6, methods for finding the phonetic transcription of OOV words will be studied.

The importance of the OOV-word problem in speech recognition systems could be significantly reduced by defining vocabularies that cover all possible user utterances. Therefore some effort will be dedicated to automatically finding vocabularies which reduce the number of expected OOV words. This will be done using information extracted from other applications (Chapter 3), and morphological information (Chapter 4).

1.3 Organization of This Thesis

I have tried to modularize the contents of the thesis as far as possible. My intention is that the reader interested in a particular chapter be able to understand it without having to read the preceding chapters. In order to achieve this, however, compactness sometimes had to be sacrificed, and certain explanations, ideas, or concepts are repeated in different chapters. For example, when reviewing previous research I restricted myself to the aspects of the publications which are directly relevant to the topic of the current chapter. As a consequence, some publications are partially reviewed in more than one chapter.

This thesis is divided into seven chapters: Four chapters are concerned with different aspects of the problem of OOV words (Chapters 3, 4, 5, and 6), and three chapters provide complementary information (Chapters 1, 2, and 7). In the following, a short summary of the contents of every chapter is given:

Chapter 1: Introduction to the OOV-word problem, approaches presented, and organization of the thesis.

Chapter 2: Review of basic speech recognition techniques, presentation of the Daimler-Benz speeech recognition system, and introduction of the evaluation measures used in the rest of the thesis.

Chapter 3: Corpus analysis of OOV words, including relationship between corpus size, vocabulary size, and OOV rate, classification of OOV words, and acoustic properties of OOV words. Study of vocabulary portability issues like cross-domain coverage and vocabulary mixing.

Chapter 4: Redefinition of the concept of "word" using automatically derived morphologically-oriented lexical units to reduce OOV rate and improve accuracy. Recovery of OOV compound words using morphological knowledge.

Chapter 5: Comparison of acoustic modeling schemes for detecting OOV words. Using garbage models for OOV words. Training the language model for OOV-word detection. Tuning it with *iterative substitution*. Study of the influence of OOV words on the system performance.

Chapter 6: Phonetically transcribing input utterances using *the cheap transcription algorithm*. Transcribing OOV words.

Chapter 7: Conclusions, contributions of this thesis, and outlook.

Finally, I should mention that throughout this thesis the first person plural is used stylistically as a synonym of the first person singular. So "we performed experiments" should be interpreted as "I performed experiments."

2 SPEECH RECOGNITION BASICS

Speech recognizers are by and large search algorithms.
Ronald Rosenfeld [Rosenfeld, 1994]

THIS chapter will first briefly discuss some standard methods and algorithms used in speech recognition. Then we will present the speech recognition system used for this thesis, stressing those parts which are unique to this system. Finally, we will summarize the quality measures used throughout the thesis.

This chapter is not intended as an introduction to speech recognition techniques, but rather as a short review of methods that are directly relevant to the content of the following chapters. The uninitiated reader is referred to introductory publications, such as the books by Lee [Lee, 1989], Huang et al. [Huang et al., 1990], and Rabiner and Juang [Rabiner and Juang, 1993].

2.1 A Generic Speech Recognition System

The task of a speech recognizer is to choose the most likely word string W^* among all possible word strings W, given the acoustic signal A:

$$W^* = \operatorname*{argmax}_W \mathrm{P}(W \mid A) \tag{2.1}$$

Using Bayes' rule we have:

$$\begin{aligned} \mathrm{P}(W \mid A) &= \frac{\mathrm{P}(A \mid W)\,\mathrm{P}(W)}{\mathrm{P}(A)} && (2.2) \\ W^* &= \operatorname*{argmax}_W \frac{\mathrm{P}(A \mid W)\,\mathrm{P}(W)}{\mathrm{P}(A)} && (2.3) \end{aligned}$$

where

$\mathrm{P}(W \mid A)$ is the probability of the word string W given the acoustic signal A, P$(A \mid W)$ is the probability that the acoustic signal A was produced by the word string W and is given by the acoustic models,

P(W) is the *a priori* probability of the word string W (given by the language model), and

P(A) is the probability of the acoustic signal A.

Since the probability of the acoustic signal P(A) is constant for all W being maximized in Equation 2.3, it can be ignored, resulting:

$$W^* = \underset{W}{\operatorname{argmax}}\, \mathrm{P}(A \mid W)\, \mathrm{P}(W) \tag{2.4}$$

Equation 2.1 requires a model for all possible acoustic signals A, whereas Equation 2.3 (or 2.4) requires a model that covers every possible word sequence W. The latter is much easier to compute because W is discrete, whereas A is a continuous signal [Ueberla, 1994]. For this reason it is easier to apply Equation 2.3 (or 2.4) to obtain the most probable word string W^*.

Equation 2.3 (or 2.4) also allows a modular strategy for speech recognition. The acoustic information contained in the speech signal is captured by the acoustic models, which provide the probability P$(A \mid W)$; and the linguistic information is given by the language model, as the probability P(W). Figure 2.1 depicts a generic speech recognizer based on this modular approach. In the following sections we will shortly review standard techniques used in such a generic system.

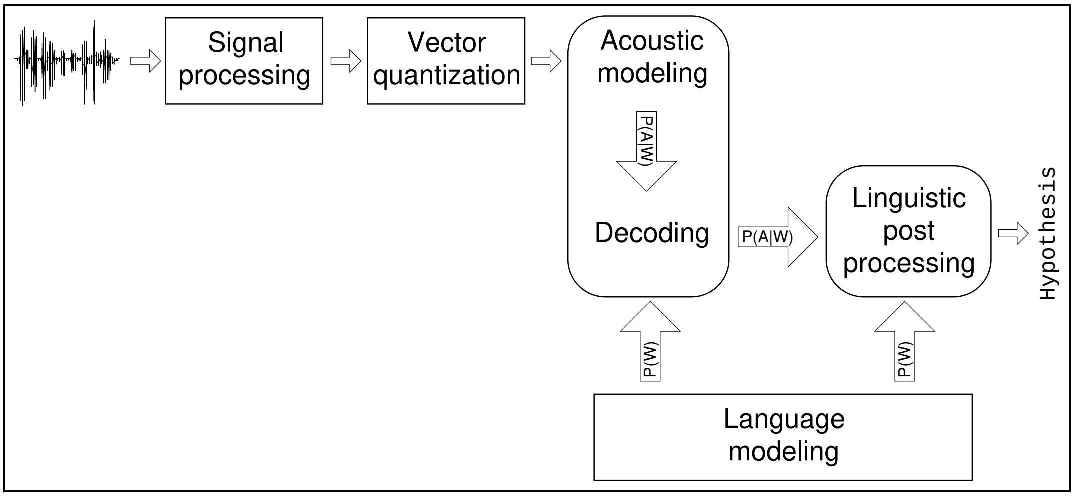


Figure 2.1: A generic speech recognition system.

2.1.1 Signal Processing

Speech is an analog signal which has to first be converted into a proper parametric representation for subsequent processing in speech recognizers. The primary goal of this parametrization is to filter out that part of the information which is relevant to determining what is being said (as opposed to information peculiar to the speaker, for instance). Another positive effect of parametrization is a great reduction in data.

The most successful techniques are Fourier analysis and Linear Predictive Coding (LPC), which can be either used alone, or in combination with cepstral analysis. These are standard signal processing methods, and will not be discussed here (see [Huang et al., 1990] for a detailed description). For the purpose of this thesis, the signal processing stage is a "black box" that transforms the speech signal into a sequence of vectors that represent the signal.

These vectors can be directly used to model the speech signal with so-called *Continuous Density Hidden Markov Models* (CDHMMs). They can also be further processed first (as explained in the next section), and then integrated with *Discrete* and *Semi-Continuous Hidden Markov Models* (SCHMMs) (see Section 2.1.3).

2.1.2 Vector Quantization

Vector quantization is used to further reduce the amount of data in the parametrized speech. The features that represents the signal can be considered as continuous, multi-dimensional vectors. If we can divide the feature space into a small number of regions (cells) spanning this entire feature space, then we can code each vector with the identification number (index) of the region it is in. This obviously is a great reduction in data over the original continuous, multi-dimensional vectors (at the cost of a certain loss of signal information).

The *codebook* consists of a set of prototype vectors that describe each cell, and its corresponding index. The codebook is created by iteratively splitting the feature space into cells and then assigning the input vectors in the training material to a given cell [Linde et al., 1980]. At the end of this process, the cell centroids and the cell indices are available. The latter are used as the alphabet of the output probabilities of discrete HMMs (see [Lee, 1989] for details).

Alternatively, each input vector is not mapped into just one prototype, but into several, and a probability is attached to each mapping. For this kind of *soft decision* vector quantization, a *probability density function* (pdf) must be estimated for each cell in the codebook. Frequently, multidimensional Gaussian distributions are chosen as pdf's. Later, the pdf's can be combined with the HMM output probabilities of semi-continuous HMMs (see [Huang et al., 1990] for details).

2.1.3 Acoustic Modeling

Hidden Markov Models (HMMs) are an efficient method of modeling the variability present in speech. An HMM consists of a collection of states and transitions. A transition probability is associated with each transition, and each state is associated with an output (or emission) probability (density) function. This structure is "traversed" as follows: Every frame the current state changes according to the transition probabilities of the state. In the new state, an output symbol is emitted based on the probability (density) function attached to the state. This way, the *temporal* variability in speech is modeled by the transition probabilities, and the *spectral* variability by the emission probabilities.

The HMM theory is usually broken down into three problems [Rabiner and Juang, 1986], [Rabiner, 1989]:

The Evaluation Problem: What is the probability that a given HMM generated a sequence of observed vectors? This problem can be solved with the *forward algorithm* (which is also used as part of the solution of the next problem).

The Training Problem: What HMM parameters maximize the probability that the HMM generates the given observation vectors? This problem is solved iteratively by means of the *forward-backward algorithm* and the *Baum-Welch re-estimation algorithm*. These algorithms are used to train the HMM transition and emission probabilities, as will be explained below.

The Decoding Problem: Which is the path through the HMM that most likely generated the observed vectors? The *Viterbi algorithm* provides an efficient method of finding the most likely path, and is therefore used for recognition (see below).

HMM Training

To train Hidden Markov Models, large amounts of speech recordings and their transliterations[1] are required. The recordings are converted into a series of feature vectors using the procedure explained in Sections 2.1.1 and 2.1.2. The transliterations are necessary to create a chain of untrained HMMs.

[1] Throughout this thesis we follow the convention used in the Verbmobil project and reserve the term "transcription" for phonetic transcription only. Orthographic transcriptions are denoted as "transliterations."

If the HMMs represent whole words, the chain of untrained HMMs can be obtained by replacing all words in each training utterance by their corresponding HMM, and then concatenating all HMMs. If the HMMs represent sub-word units, such as triphones[2], all words are first mapped into their phonetic representation (using a pronunciation dictionary), then into a succession of triphones, and finally into the untrained HMM chain. In both cases, the topology (structure) of the HMMs must be defined in anticipation.

The untrained HMM chain can be first initialized using time labeled training material (where the temporal boundaries of the phones or triphones are signaled). This step is optional, though, since the forward-backward algorithm both aligns the HMM chains with the feature vectors and then trains them.

Slightly different versions of the Baum-Welch algorithm are used to train discrete and semi-continuous HMMs, and continuous density HMMs (see [Huang et al., 1990]). In all three cases, after training a set of transition and emission probabilities is available for each HMM state.

Decoding

Given a spoken utterance and a set of trained HMMs that represent this utterance, the most likely spoken sequence can be effectively found using the Viterbi algorithm. For this, a network containing the HMMs of all words in the vocabulary (or the concatenation of their sub-word units) is necessary. This structure is called the *recognition lexicon*. Two frequent ways of organizing such a lexicon are as a linear structure or as a tree. In the former, the HMMs of each word in the vocabulary are stored separately: For a vocabulary of V words, the lexicon consists of V parallel paths. In a lexicon tree, the left parts of two or more words having the same HMM structure are "factored out." In this case, the lexicon has only one root node and V final nodes (the leaves).

The Viterbi algorithm is a time-synchronous algorithm that traverses the lexicon as follows: At any given time t in the search, one or more HMM states s_t^i are considered, where i is the node being passed through at time t. For every state s_t^i at time t, only the path to s_t^i which was most probable is remembered along with its probability. At word ends, the transition probability is replaced by the probability given by the language model, as will be explained below. At the end of the process, the highest probability among all final states is determined, and the most likely state sequence can be back-traced from $t = T$ (final time) to $t = 0$.

Usually, the Viterbi algorithm is combined with the *beam search* to reduce the search space. At every time frame the path with the highest probability P^* is

[2] A sub-word unit consisting of a single phone within a specific left and right context is called a *triphone*.

identified, and all paths having a probability lower than a given fraction of P^* are discarded.

Various modifications of the Viterbi algorithm have been proposed, which aim at generating not only the most likely (best) hypothesis, but also several alternatives (see, for example, [Kaltenmeier, 1991], [Schwartz and Austin, 1991], and [Oerder and Ney, 1993]). The most common structures outputted by the modified algorithms are: N-best sentences, word lattices, and word graphs.

N-best is a list of the N distinct sentences with the highest (Viterbi or forward) probabilities, along with their probabilities.

Word Lattices are meshed structures of nodes and arcs. Arcs contain words labels and their probabilities, and nodes denote the temporal boundaries of the words. Lattices might contain gaps and overlaps, and might be seen as non-optimized word graphs.

Word Graphs are word lattices which have been optimized using heuristic criteria. As word lattices, they also contain arcs and nodes that represent word labels, word probabilities, and word boundaries. (See Figure 2.5 for an example of a word graph.)

2.1.4 Linguistic Post-Processing

If the Viterbi algorithm produces the best sentence, this sentence is the output of the recognizer. On the other hand, if alternate hypotheses are generated, other system modules have to find the best hypothesis among the candidates.

The idea behind this is to apply different knowledge sources at different stages during recognition, thus making the recognition process more modular and requiring less computation. In the first stage, a preliminary set of alternative candidates is produced. These candidates are then further processed using higher-level knowledge sources, such as language models. From this point of view, recognition can be viewed as sieving information over several refinement levels. The best hypothesis can be found with various methods, including modified forward and Viterbi searches, and dynamic programming methods (see, for example [Murveit et al., 1993] and [Kuhn et al., 1996]).

2.1.5 Language Modeling

The function of a language model is to provide the speech recognizer with the *a priori* probability $P(W)$ of a hypothesized word sequence W. $P(W)$ can be decomposed into the product of the word probabilities that make up the word sequence W:

$$\mathrm{P}(W) = \mathrm{P}(w_1^n) = \prod_{i=1}^{n} \mathrm{P}(w_i \mid h_i) \tag{2.5}$$

where w_i is the word in question, and h_i its *history*, that is, its preceding context (w_1^{i-1}). Equation 2.5 is very convenient, because it allows $\mathrm{P}(W)$ to be computed sequentially. It also allows to concentrate on its constituent factors, $\mathrm{P}(w_i \mid h_i)$, which can from now on be analyzed independently of $\mathrm{P}(W)$. Finally, it introduces the concept of history, which is of utmost importance in language modeling: The amount and type of history considered characterize the model in question. The most frequently used language models will now be discussed.

If the history is reduced to the $N-1$ preceding words, the resulting model is an N-gram model and, $\mathrm{P}(w_i \mid h_i)$ is estimated as follows:

$$\mathrm{P}(w_i \mid h_i) \approx \mathrm{P}(w_i \mid w_{i-N+1}^{i-1}) \tag{2.6}$$

To choose the value of N, many factors must be taken into account. Notably, the amount of training data available and the size of the vocabulary strongly constrain the value of N: For a vocabulary of V words, V^N probabilities have to be estimated during training, making models with $N > 3$ very difficult to train appropriately (because many word sequences are not observed at all; see Equation 2.10). In addition, interfacing the language model to the decoder becomes increasingly more resource-consuming for longer histories. Therefore, N-gram language models for speech recognition are usually either trigrams ($N = 3$) or bigrams ($N = 2$).

A variation of these N-gram language models are the class-based N-gram language models. Instead of using exclusively the *identity of the words* contained in the history, class-based language models also make use of their *word classes*. Word class information can be incorporated into the N-gram scenario in various ways. The most obvious ones include:

$$\mathrm{P}(w_i \mid h_i) \approx \mathrm{P}(w_i \mid \mathrm{g}(w_{i-N+1}^{i-1})) \tag{2.7}$$

$$\mathrm{P}(w_i \mid h_i) \approx \mathrm{P}(w_i \mid \mathrm{g}(w_i))\, \mathrm{P}(\mathrm{g}(w_i) \mid \mathrm{g}(w_{i-N+1}^{i-1})) \tag{2.8}$$

where $\mathrm{g}(w)$ is a function that maps a word w into its class. Class-based language models have less free parameters than word-based N-grams, provided that the number of classes G is smaller than the vocabulary size V. Equation 2.8, for example, requires estimating $G^N + GV$ probabilities—a significant reduction compared to the V^N free parameters of an N-gram language model. Therefore, class-based language models are especially suitable whenever the amount of training data available is scarce, but tend to perform worse than word-based N-grams if large amounts of data are available [Ueberla, 1995].

Other ways of taking advantage of the information conveyed in the history of a word have also been investigated, and resulted in other language modeling paradigms. Among them are decision trees, caches, and maximum entropy models, to name but a few. For a more thorough discussion of existing language models, see for example [Rosenfeld, 1994] or [Ueberla, 1994].

We will now focus our attention on N-gram language models, and in particular, on the estimation of its N-gram probabilities. The N-gram probabilities, and the conditional probabilities of a word given the $N-1$ preceding words can be estimated using the relative frequency of occurrence of the N-grams in the training corpus of size C, as follows:

$$\mathrm{P}(w_{i-N+1}^{i}) = \frac{\mathrm{Count}(w_{i-N+1}^{i})}{C} \tag{2.9}$$

$$\mathrm{P}(w_i \mid w_{i-N+1}^{i-1}) = \begin{cases} \frac{\mathrm{Count}(w_{i-N+1}^{i})}{\mathrm{Count}(w_{i-N+1}^{i-1})} & \text{if} \quad \mathrm{Count}(w_{i-N+1}^{i-1}) > 0 \\ 0 & \text{if} \quad \mathrm{Count}(w_{i-N+1}^{i-1}) = 0 \end{cases} \tag{2.10}$$

where the function Count returns the frequency of a word (or word sequence) in an underlying text.

This method is called *Maximum Likelihood Estimation, MLE*. From Equation 2.9 it is evident that MLE assigns a probability of zero to all N-grams which have not been observed in the training material. This is not desired in most applications of speech recognition dealing with large vocabularies and spontaneous speech. To avoid zero probabilities, *smoothing* techniques are employed. Smoothing consists of redistributing probability mass from the observed N-grams to the N-grams which have not occurred in the training corpus.

The amount of probability mass to be redistributed can be gauged manually, or estimated from the training data. Good [Good, 1953] used the proportion of events (in our case N-grams) which occurred only once as an estimate for the probability of unseen events. This probability mass is discounted from rare events (in our case, N-grams with low counts). The precise formulation of the so-called Good-Turing discounting strategy can be found, for example in [Clarkson and Rosenfeld, 1997, Equation (2)]. There, other discounting methods are also discussed.

An alternative to smoothing is to rely in lower-order estimates. Whenever an N-gram probability is zero, the (N-1)-gram probability is considered. This is called "backing off." The language models used in this thesis are based on a combination of Good-Turing discounting and backing off, as introduced in [Katz, 1987]. This will be discussed in more detail in Section 5.4.

2.2 The Daimler-Benz Speech Recognition System

A modular system with a uniform architecture to handle many different tasks was developed at the Daimler-Benz Research Center. The features of the system include both isolated word and continuous speech recognition, flexible vocabulary, environmental robustness, and ability to cope with spontaneously spoken speech.

In the following sections we will describe the particularities of this system. We will concentrate on aspects directly related to the subject of this thesis. Other system modules will only be briefly mentioned. The reader is referred to more extensive discussions in the following publications: [Class et al., 1993a] (signal processing), [Class et al., 1992] (vector quantization), [Kaltenmeier, 1991] and [Class et al., 1993b] (HMM training), [Kuhn et al., 1996] (decoding), [Brietzmann et al., 1994] and [Kuhn et al., 1996] (linguistic post-processor).

2.2.1 Signal Processing

The input to the recognizer is an audio signal, which has been sampled and quantized. The sampling rate and resolution depend on the application. In VERBMOBIL (see Section 3.3.2), for instance, 16 kHz and 16 bits are used, respectively. Each frame containing 2^9 samples is first fed to a Hamming window and then converted by an FFT into a spectral representation. This discrete spectrum is filtered using the Mel scale and finally cosine transformed into the cepstral domain. The coefficients with the highest variances are chosen to represent the signal. Figure 2.2 shows a block diagram of this feature extraction stage.

Using this procedure, feature vectors are produced at a rate of one frame per centisecond. A feature vector consists of 13 coefficients: 12 cepstral coefficients and 1 logarithmic energy component. The cepstral features are then normalized to remove spectral influences of the speaker, microphone, room acoustics, and transmission line. This normalization can be interpreted as an unsupervised speaker and system adaptation [Class et al., 1993b].

For *training* purposes only (and not for recognition), Δ and $\Delta\Delta$ features [Furui, 1986] are computed from the normalized cepstral coefficients and the logarithmic energy contour, using a window of 5 frames. This yields a total number of 39 parameters per frame.

2.2.2 Vector Quantization

We use soft-decision vector quantization with Gaussians distributions and full covariance matrices. The codebook is created using the algorithm described in [Linde et al., 1980]. The output of the vector quantizer is a number of likelihoods

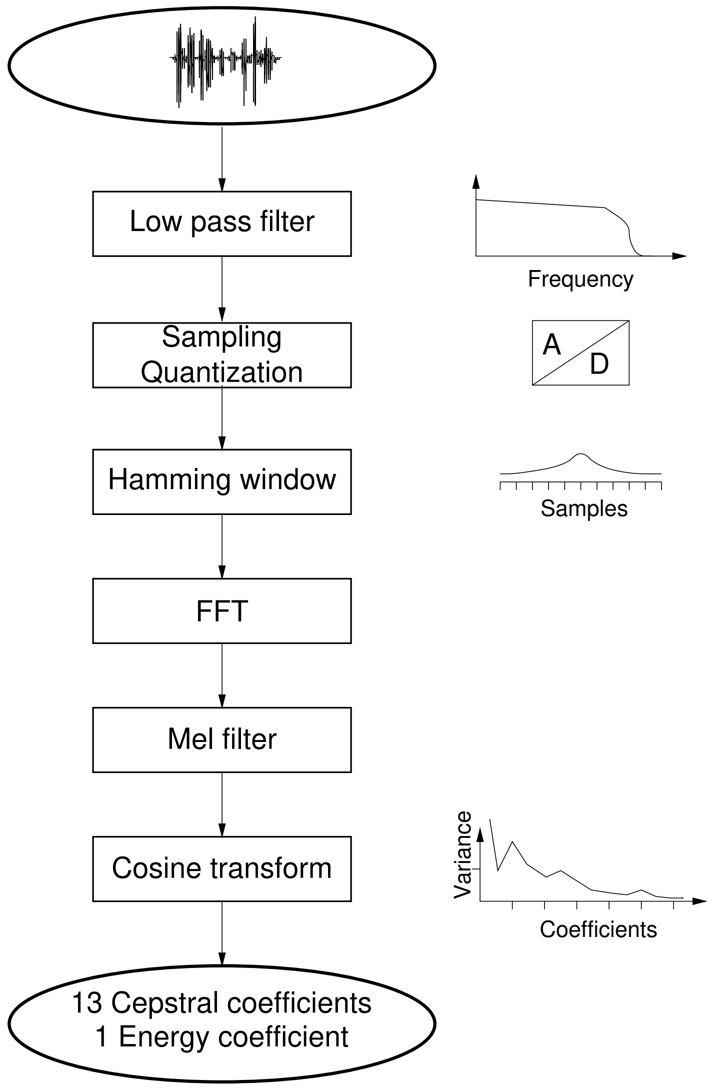


Figure 2.2: The signal processing module in the Daimler-Benz speech recognition system.

with their corresponding class indices for each input feature vector (see [Class et al., 1991, Class et al., 1992] for more details). Different codebooks are employed for training and recognition:

Training: Each feature vector containing 39 coefficients is vector quantized with multiple codebooks: for cepstral, log energy, Δ, and $\Delta\Delta$ coefficients. These codebooks consist of either 2^8 Gaussian distributions for each the cepstral, Δ, and $\Delta\Delta$ coefficient, or 2^4 Gaussian distributions for the energy.

Recognition: The so-called *LDA codebook* (see below) is employed, which has 32 dimensions and 2^{10} Gaussian distributions.

2.2.3 Acoustic Modeling

HMM Training

With the four data streams produced by the vector quantizer (cepstral, log energy, Δ, and $\Delta\Delta$), a first set of HMMs is trained using the forward-backward algorithm. First, context-independent phone models are trained. Then, context-dependent models are initialized with the parameters of the context-independent models and then trained. During the last iteration of this training step, the mean values and covariances are computed, which are required for the following stage.

Linear Discriminant Analysis. The Linear Discriminant Analysis (LDA) is an efficient method for separating classes in the feature space [Fukunaga, 1972]. The problem of implementing LDA in an HMM recognizer based on sub-word units is defining the classes and computing the corresponding scatter (covariance) matrices [Class et al., 1993b]. At the Daimler-Benz Research Center, this problem was solved as follows: Classes are directly assigned to semi-continuous HMM states or state clusters of sub-word units. And covariance matrices are computed during the forward-backward training mentioned above. Using these class-specific means and covariances, the LDA transformation matrix is computed.

The LDA module can then be fed with feature vectors. The cepstral and energy coefficients of several frames adjacent to the current frame are used. Since these time windows implicitly include temporal dynamic information, Δ and $\Delta\Delta$ coefficients are no longer utilized [Kaltenmeier, 1991]. In the system used in this thesis, 9 frames are added into one $9 \times 13 = 117$-dimensional super-vector which is transformed into 32 coefficients by the LDA transformation matrix. The classic way of applying LDA is to compute a new codebook using LDA-transformed training samples. This is simplified by directly transforming the codebook mean values and covariances by the LDA transformation matrix. This way, a codebook containing 32-dimensional mean values and covariances is obtained, which was dubbed the *LDA codebook* [Class et al., 1993b].

After all training samples have been LDA-transformed and vector quantized with the LDA codebook, the next training stage can be initiated. There, the context-independent phone models are trained again using LDA-transformed input data, and finally, the context-dependent polyphone models are trained, initialized with the parameters of the context-independent models.

The training method can be summarized in the following steps:

1. Train simple models, then
2. train complex models, and then
3. use them to initialize a new training.
4. Train simple models again, and finally
5. train complex models again.

Figure 2.3 gives an overview of the complete training procedure.

The context-dependent models used are polyphones. Polyphones are phones in a generalized context, whose length depend on the availability of training material. The more training samples available, the longer the context[3]. Also, very frequent words are modeled separately with whole-word models. The set of polyphones and whole-word models is defined by an automatic procedure, which is run before the HMM training begins [Fach, 1996].

The topology of the models is rather standard: Every state is connected to the immediately adjacent state to its right and to itself by a (loop) transition. Some states can be skipped when the HMMs are traversed. The number of states in the model depends on the length of the units being modeled. For polyphones, for instance, three states are used, whereas for whole-word models, between five and fifteen states are more appropriate.

Decoding

The Daimler-Benz speech recognizer used in this thesis is a two-stage system. First, a large word graph is generated using a modified Viterbi algorithm. This word graph is then further processed and reduced by a linguistic postprocessor (see Section 2.2.4), which incorporates higher-level knowledge sources. Figure 2.4 depicts the two-stage recognition approach.

[3]We refer to context within words. Context across words is not considered.

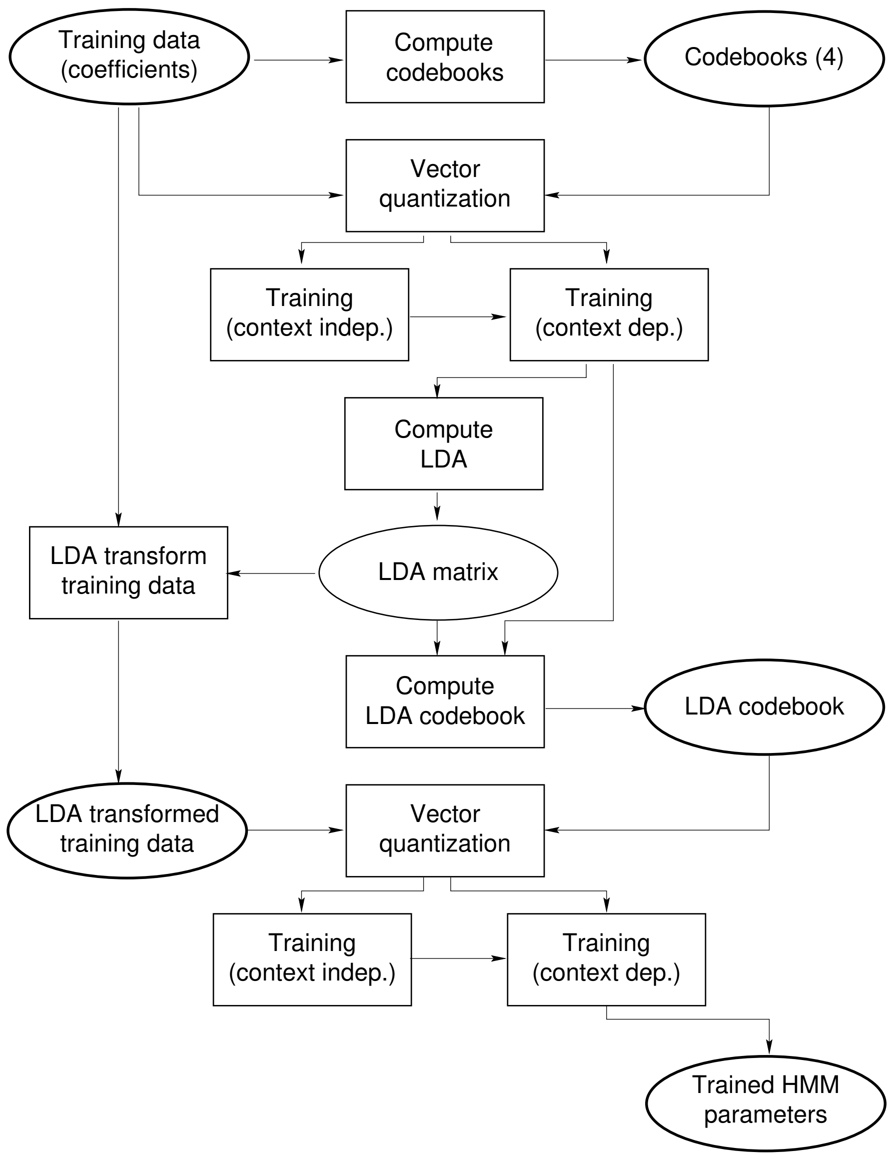


Figure 2.3: Overview of the HMM training in the Daimler-Benz speech recognition system.

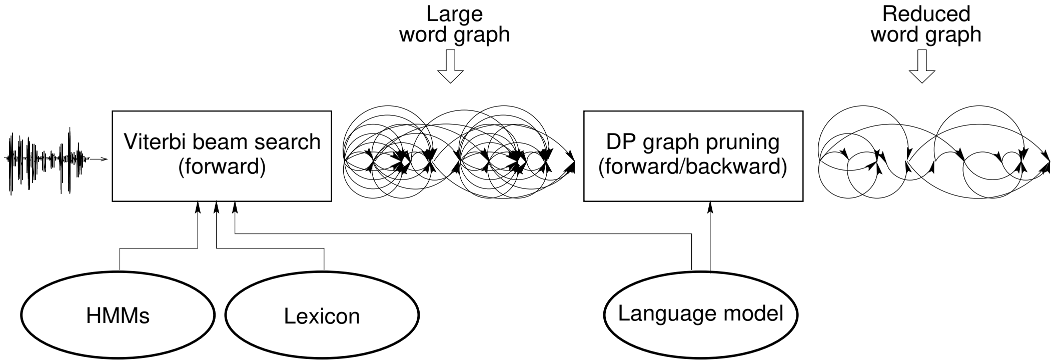


Figure 2.4: Two-stage architecture of the Daimler-Benz speech recognition system.

The first stage of the recognizer is based on the Viterbi algorithm. Here, at least three parameter sets are needed: the HMMs trained as explained above, a lexicon, and a bigram language model. The lexicon is compiled off-line, and is implemented as a tree for efficiency reasons. Every node is an HMM state, and every leaf, a word of the recognition vocabulary. The language model is also trained off-line, as will be explained in Section 2.2.5.

During the search, the lexicon is traversed, and all active paths are expanded in every frame. In order to survive, a path must have a greater probability than the pruning probability. At word ends, a bigram language model probability to the succeeding word is estimated among possible successors. Word hypotheses are incorporated into a word-graph structure asynchronously, provided that they fulfill various prerequisites, such as having a minimal duration, being among the n most probable words in the current frame, etc.

The word graphs used at Daimler-Benz are directed acyclic graphs, where every edge g_i contains the following information:

$$g_i = \{a(i),\ e(i),\ w_i,\ t_a(i),\ t_e(i)\ , s(i)\}$$

$a(i)$ and $e(i)$ denote the start and end nodes of word w_i, $t_{a(i)}$ and $t_{e(i)}$ their corresponding time frames, and $s(i)$ the logarithmic HMM score of the word w_i in the region of $[t_a(i), t_e(i)]$. Figure 2.5 shows an example of a small word graph. In practice, very large word graphs, with approximately 100 hypotheses per spoken word are produced. This large structure is kept in memory, and passed over to the linguistic postprocessor at the end of the Viterbi search.

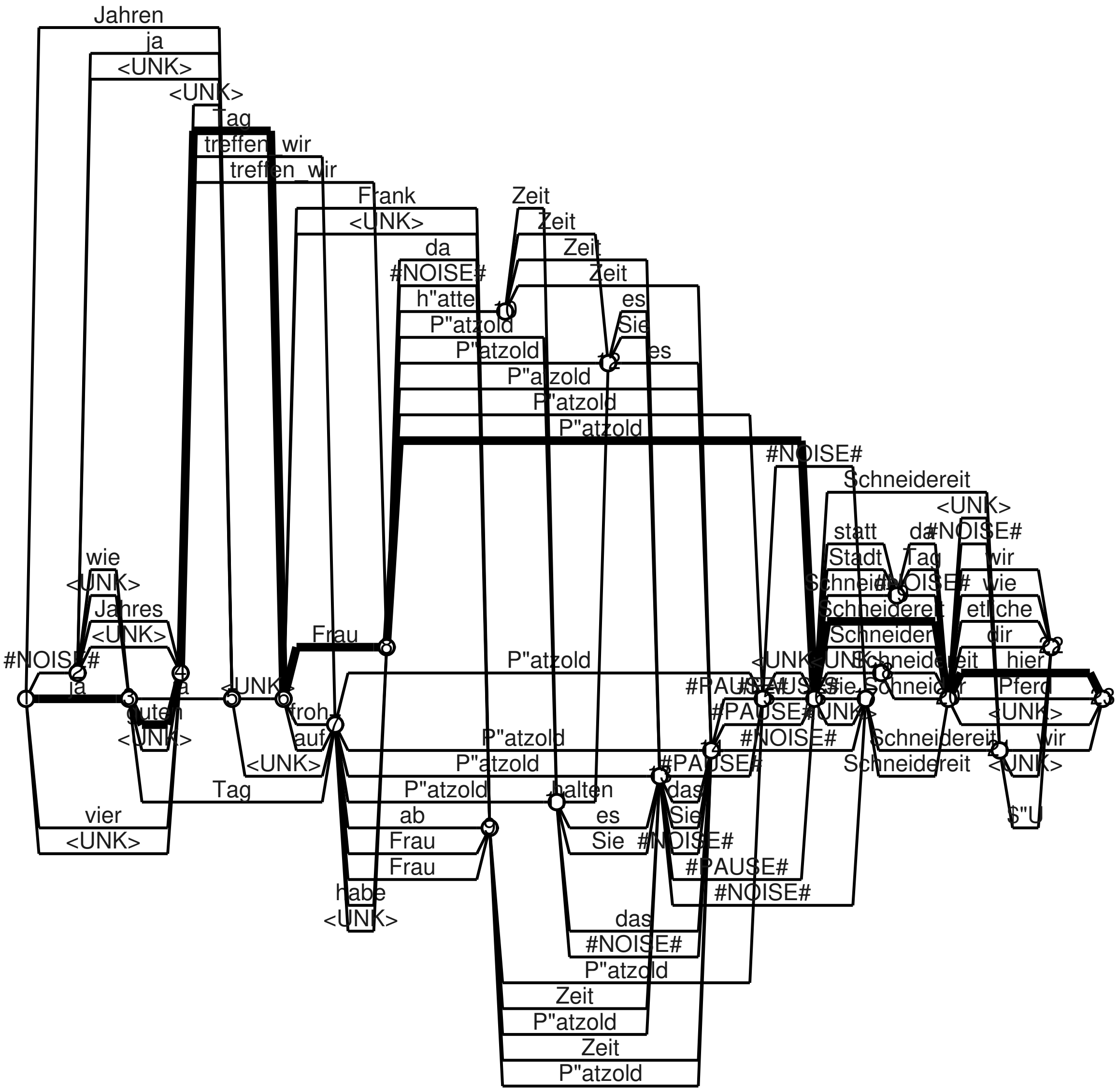


Figure 2.5: Example of a word graph hypothesized by the Daimler-Benz speech recognizer. For better visualization, only the word labels were attached to the edges (and not the scores and time information). The German sentence *"ja guten Tag Frau Pätzold Schneidereit hier"* (hello Ms. Pätzold Schneidereit speaking) was uttered. The bold path through the graph shows this sentence.

2.2.4 Linguistic Post-Processing

Using a dynamic programming method introduced in [Kuhn et al., 1996], the word graph produced by the Viterbi decoder can be reduced either to a smaller word graph (as shown in Figure 2.4), or to the first-best sentence. This method integrates higher-level knowledge sources into the recognition process. To simplify the explanation, we will consider the integration of a bigram language model only.

For each edge g_i in the word graph generated by the decoder, the score of the best path which passes through g_i is computed by a forward and a backward recursion. The forward path is computed from left to right as follows:

$$\alpha_{e(i)}(i) = s(i) + \min_{i-1:\, a(i)=e(i-1)} (\alpha_{e(i-1)}(i-1) + \vartheta \, \log(\mathrm{P}(w_i \mid w_{i-1}))) \qquad (2.11)$$

This formula is used to update α from the end of the previous word w_{i-1}, to the end of the current word w_i by adding[4] the acoustic score $s(i)$ and the language model score $\vartheta \, \log(\mathrm{P}(w_i \mid w_{i-1}))$. The factor ϑ weights the language model score and is optimized empirically.

A similar recursion has to be computed for the backward path $\beta_{a(i)}(i)$, but as opposed to α, the β values are indexed by the end node of the edge g_i:

$$\beta_{a(i)}(i) = s(i) + \min_{i+1:\, e(i)=a(i+1)} (\beta_{a(i+1)}(i+1) + \vartheta \, \log(\mathrm{P}(w_{i+1} \mid w_i))) \qquad (2.12)$$

During pruning, only those edges g_i, g_{i+1} having a score greater than or equal to a given fraction of the best path in the word graph are kept:

$$\alpha_{e(i)}(i) + \beta_{a(i+1)}(i+1) + \vartheta \, \log(\mathrm{P}(w_{i+1} \mid w_i)) \geq \theta \, \alpha^* \quad \text{with} \quad e(j) = a(i) \qquad (2.13)$$

and the rest are discarded. α^* denotes the best path determined by the forward recursion, and θ the pruning threshold. The pruning threshold determines the density of the resulting graph, and must be given as a parameter for recognition.

2.2.5 Language Modeling

Several different language model interfaces are integrated in the Daimler-Benz speech recognizer. For this thesis, only back-off N-grams were employed: Bigrams were used with the Viterbi decoder, and bigrams or trigrams with the linguistic postprocessor. Both bigrams and trigrams were trained using the CMU Statistical Language Modeling toolkit described in [Rosenfeld, 1995a]. In Section 5.4 back-off language models will be discussed in more detail.

[4] Note that since all scores concerned are logarithmic (and negative), additions correspond to a multiplicative accumulation of scores.

2.3 Usual Measures in Speech Recognition

In this section we briefly review the quality measures used throughout this thesis. Since all measures are well established, we will not discuss their pros and cons. Their limitations will be mentioned whenever pertinent. A few more specific quality measures are used in a few chapters. In these cases, they will be introduced at the appropriate place.

2.3.1 Evaluating Speech Recognition Performance

Word Error Rate, Word Accuracy, and Word-Correct Rate

Word error rate and word accuracy are the standard measures of goodness of a speech recognizer. They are computed by first aligning the spoken reference string of text with the hypotheses of the recognizer. The alignment is achieved using a search based on dynamic programming, during which a cost function is minimized. The cost function usually chosen is the Levenshtein distance, which weights word insertions, substitutions and deletions equally. For a detailed explanation of the alignment algorithm, see [Picone et al., 1990], [Picone et al., 1986], or [Lehning, 1994b]. Figure 2.6 shows an example of such an alignment.

Reference:	this	is	a	bad	speech	recognizer	
Hypothesis:	this	—	about		speech	wreck	nice
Alignment[5]:	`COR`	`DEL`	`DEL`	`SUB`	`COR`	`SUB`	`INS`

Figure 2.6: Example of an alignment between the spoken (Reference) string and the recognized hypothesis.

A few words should be said about how German homonyms[6] are handled in the VERBMOBIL project (see Section 3.3.2), which we adopted for this thesis. During the alignment, confusions between homonyms with different spelling, as in the English words `their` ↔ `there`, are counted as errors. Spelling differences that concern only the case of the words involved, as in the German `gehen` ↔ `Gehen` (*verb, to go* ↔ *noun, the act of going*), are considered correct recognitions. This apparent inconsistency is due to the nature of the German orthography, where all verb infinitives can be used as nouns, which are always capitalized in German.

[5]The alignment between the strings *"is a bad"* and *"about"* could be `DEL DEL SUB`, or `DEL SUB DEL`, or any other permutation that keeps the number of errors constant.

[6]Homonyms are words with the same sound but different meanings.

After the alignment is completed, the number of errors, that is insertions, substitutions and deletions, is counted. The *Error Rate* (ER) can be calculated as the sum of the insertion (INS), substitution (SUB), and deletion (DEL) rates:

$$\mathrm{ER} = \frac{\mathrm{Count}(\texttt{INS})}{\mathrm{Count}(\texttt{REF})} + \frac{\mathrm{Count}(\texttt{SUB})}{\mathrm{Count}(\texttt{REF})} + \frac{\mathrm{Count}(\texttt{DEL})}{\mathrm{Count}(\texttt{REF})} \tag{2.14}$$

where Count(REF) denotes the total number of words spoken. The *Word Accuracy* (WA) is simply the complement of the error rate:

$$\begin{aligned} \mathrm{WA} &= 1 - \mathrm{ER} \\ \mathrm{WA} &= 1 - \frac{\mathrm{Count}(\texttt{INS}) + \mathrm{Count}(\texttt{SUB}) + \mathrm{Count}(\texttt{DEL})}{\mathrm{Count}(\texttt{REF})} \end{aligned} \tag{2.15}$$

The *Word-Correct rate* (WC) is similar to the word accuracy, but here insertions are not counted as errors:

$$\mathrm{WC} = 1 - \frac{\mathrm{Count}(\texttt{SUB}) + \mathrm{Count}(\texttt{DEL})}{\mathrm{Count}(\texttt{REF})} \tag{2.16}$$

We will *not* report results using the word-correct rate in this thesis. It was presented at this point just to emphasize this fact. We will rather use indistinctively either word accuracy or word error rate.

Until now, we assumed that the recognizer output is a single string. If the recognizer produces word graphs instead, the path with the smallest Levenshtein distance in the graph is aligned with the reference to compute word accuracy. Since graphs with a larger number of alternatives per spoken word tend to contain fewer errors, it is indispensable to specify the number of hypotheses per spoken word, or graph density, along with the word accuracy to compare recognition results. They can be represented in plots with the graph density on the abscissa and the word accuracy on the ordinate (see, for example, Figure 4.9 on page 75).

After studying the exact formulations of Equations 2.14 through 2.16, the reader may have the impression that the measures presented are exact quantities. Actually, they are estimated values. Around the (unknown) *exact* value of a random variable v is an interval within which its *estimated* value $\hat{v}$ is situated with a given probability γ. This interval is called the confidence interval for the probability γ. The confidence interval width (which in turn is also a random variable) can be expressed in terms of the amount of data used to estimate $\hat{v}$, and the estimated value $\hat{v}$ itself. In the case of word accuracy, smaller accuracies have wider confidence intervals. Also, the more test data is used, the narrower the confidence interval becomes. In VERBMOBIL, the size of the different test sets used was chosen so that the resulting 95 % confidence intervals be below 1 % word accuracy [Paulus and Lehning, 1994].

Test Set Perplexity

The word accuracy is the ultimate measure of performance in speech recognition. Even so, it is desirable to optimize the modules involved in the recognition process separately, following the principle of "divide and conquer." The measure used for this separate optimization, though, should be highly correlated with the word accuracy (more on this below). For language modeling, there is such a measure, and it is called *perplexity*.

If we have access to the source that "produces" the language, the perplexity can be computed exactly from the entropy of the source [Lee, 1989, Appendix I]. This is the case, for instance, when the language model is a finite state grammar. If the language source is not available explicitly, the perplexity can only be estimated using a test set. This is the so-called *test set perplexity*, which asymptotically approaches the exact perplexity for large test sets. Since in this thesis we deal exclusively with stochastic language models, we will use only the test set perplexity. For simplicity, though, we will frequently refer to it as simply "perplexity."

The test set perplexity measures the quality of the language model. Since language models are intended to provide an accurate estimate of the probability of a given word sequence, it is intuitive to involve these probabilities when computing perplexity. The total probability of a word sequence $w_1^n = w_1, w_2, \ldots, w_n$ is[7]:

$$\mathrm{P}(w_1^n) = \prod_{i=1}^{n} \mathrm{P}(w_i \mid w_1^{i-1}) \tag{2.17}$$

and the perplexity is simply the reciprocal of the geometric mean of the total probability:

$$\mathrm{PP} = \left(\prod_{i=1}^{n} \mathrm{P}(w_i \mid w_1^{i-1}) \right)^{-1/n} \tag{2.18}$$

Since for long word sequences, $\mathrm{P}(w_1^n)$ can become very small, logarithmic scaling is normally used. In this case, the perplexity can be computed as follows:

$$\begin{aligned} \log_2(\mathrm{PP}) &= -\frac{1}{n} \sum_{i=1}^{n} \log_2(\mathrm{P}(w_i \mid w_1^{i-1})^{-1/n}) \\ \mathrm{PP} &= 2^{-\frac{1}{n} \sum_{i=1}^{n} \log_2(\mathrm{P}(w_i|w_1^{i-1}))} \end{aligned} \tag{2.19}$$

Thus, when an N-gram language model is used, the N-gram test set perplexity is expressed as:

$$\mathrm{PP} = 2^{-\frac{1}{n} \sum_{i=1}^{n} \log_2(\mathrm{P}(w_i|w_{i-N+1}^{i-1}))} \tag{2.20}$$

[7]The conditional probability of the first word given its context is defined as: $\mathrm{P}(w_1 \mid w_1^0) = \mathrm{P}(w_1)$.

From Equations 2.19 and 2.20 the following interpretations of perplexity can be derived: Perplexity represents the average ramification or branching factor of the language (as given by the model). Alternatively, it represents the degree of "surprise" contained in the word sequence w_1^n. One can imagine the speech recognizer trying to decide on the identity of next word and consulting the language model. The number of candidates returned[8] on the average by the language model will be PP. Viewed this way, the function of the language model is reducing the vocabulary "perceived" by the recognizer. If no language model is used, the perceived vocabulary size is V. In this case, the perplexity is the highest possible, namely V, because all words are equally probable. If a language model of perplexity PP is used, the perceived vocabulary size will be PP such that $PP < V$.

It is important to emphasize that the test set perplexity depends not only on the language model, but also on the test set itself. In order to fairly compare two different language models, exactly the same test set must be used. In addition, the vocabulary of the models compared should also be the same. The perplexity tend to be lower with smaller vocabularies, biasing the comparison results towards models with smaller vocabularies.

It was mentioned above that perplexity and word error rate are highly correlated. Even though there is no known analytical expression that relates these parameters, a strong regularity between them has been empirically observed. This is known as the "square-root" rule of thumb. When comparing two language models of different perplexities PP_2 and PP_1 (with $\mathrm{PP}_2 < \mathrm{PP}_1$), the improvement in error rate is approximately equal to the square root of the relative decrease in perplexity:

$$\begin{aligned} \sqrt{1 - \frac{\mathrm{PP}_1 - \mathrm{PP}_2}{\mathrm{PP}_1}} &\approx 1 - \frac{\mathrm{ER}_1 - \mathrm{ER}_2}{\mathrm{ER}_1} \\ \sqrt{\frac{\mathrm{PP}_2}{\mathrm{PP}_1}} &\approx \frac{\mathrm{ER}_2}{\mathrm{ER}_1}; \quad \text{for} \quad \mathrm{PP}_2 < \mathrm{PP}_1 \end{aligned} \tag{2.21}$$

[8]This is an oversimplification of the more complicated true picture. Candidates are not "returned" by the language model. The decision on the most likely candidate is made taking the acoustic scores together with the linguistic scores provided by the language model into account, as explained elsewhere in this chapter.

2.3.2 Evaluating Out-Of-Vocabulary Words

Out-Of-Vocabulary Rate

The proportion of Out-Of-Vocabulary words in a given reference text and for a particular vocabulary is measured by the *Out-Of-Vocabulary rate* (OOV rate):

$$\text{OOV rate} = \frac{\text{Count}(\texttt{OOV}_{\texttt{REF}})}{\text{Count}(\texttt{REF})} \tag{2.22}$$

where $\text{Count}(\texttt{OOV}_{\texttt{REF}})$ denotes the number of OOV words in the reference text, and $\text{Count}(\texttt{REF})$ the total number of words.

Recall and Precision

The quality of OOV-word detection is expressed in the two parameters *Recall* (RCL) and *Precision* (PRC). Recall is the ratio of the number of correctly detected OOV words to the total number of spoken OOV words in the reference text, and precision is the ratio of correctly detected OOV words to the number of hypothesized OOV words:

$$\text{RCL} = \frac{\text{Count}(\texttt{OOV}_{\texttt{COR}})}{\text{Count}(\texttt{OOV}_{\texttt{REF}})} \tag{2.23}$$

$$\text{PRC} = \frac{\text{Count}(\texttt{OOV}_{\texttt{COR}})}{\text{Count}(\texttt{OOV}_{\texttt{HYP}})} \tag{2.24}$$

Recall depicts the proportion of all OOV words which were detected as such. Precision measures the proportion of hypothesized OOV which were correct “hits.” If all OOV-word hypotheses are correct, for instance, the precision will be 100%. If, on the other hand, no hypothesis is correct, the precision will be 0%. As can be clearly seen from Equation 2.24, precision is not defined if no OOV words are hypothesized.

There are also other alternative measures of the quality of OOV-word detection. The most wide-spread are *Detection Rate* (DR) and *False Alarm Rate* (FAR), which are especially used in the field of keyword spotting (see Section 5.2.3). The detection rate is equivalent to recall, and is also calculated with Equation 2.23. The false alarm rate is comparable with the precision in the sense that both register the number of wrongly hypothesized OOV words. There are different ways of defining the false alarm rate. To our understanding, the most convenient definition for the purpose of OOV detection is the following: The false alarm rate is the quotient of the number of OOV words incorrectly hypothesized and the total number of OOV words in the reference. The following equations show the conversion of recall and precision to detection and false alarm rates:

$$\mathrm{DR} = \mathrm{RCL} \tag{2.25}$$

$$\mathrm{FAR} = \frac{\mathrm{RCL}}{\mathrm{PRC}} - \mathrm{RCL}; \quad \text{for} \quad \mathtt{OOV}_{\mathtt{COR}} > 0 \tag{2.26}$$

One disadvantage of both recall and precision, and detection rate and false alarm rate is that they always have to be reported in pairs. It would be very appealing to have a parameter that captures both measures in just one number. Our VERBMOBIL project partners from the University of Bielefeld proposed such a measure and dubbed it *Detection Accuracy* (DA) [Jusek et al., 1995] which is defined as follows:

$$\begin{aligned} \mathrm{DA} &= \frac{\mathrm{Count}(\mathtt{OOV}_{\mathtt{COR}}) - \mathrm{Count}(\mathtt{OOV}_{\mathtt{INC}})}{\mathrm{Count}(\mathtt{OOV}_{\mathtt{REF}})} \\ &= \frac{2\,\mathrm{Count}(\mathtt{OOV}_{\mathtt{COR}}) - \mathrm{Count}(\mathtt{OOV}_{\mathtt{HYP}})}{\mathrm{Count}(\mathtt{OOV}_{\mathtt{REF}})} \end{aligned} \tag{2.27}$$

where $\mathrm{Count}(\mathtt{OOV}_{\mathtt{COR}})$ is, as before, the number of OOV words correctly detected, and $\mathrm{Count}(\mathtt{OOV}_{\mathtt{INC}})$ is the number of OOV words incorrectly hypothesized. Both counts have to be calculated on the alignment (see Section 2.3.1). This fact constitutes the main fallacy of the detection accuracy. Whenever the number of incorrectly hypothesized OOV words that appear in the alignment is not equal to the total number of incorrectly hypothesized OOV, the detection accuracy will provide misleading figures. This is the case when evaluating word graphs.

For this reason, detection accuracy will not be used to report results in this thesis. In any case, detection accuracy can be calculated from recall and precision as follows:

$$\mathrm{DA} = 2\,\mathrm{RCL} - \frac{\mathrm{RCL}}{\mathrm{PRC}}; \quad \text{for} \quad \mathtt{OOV}_{\mathtt{COR}} > 0 \tag{2.28}$$

provided that they were computed on a first-best hypothesis, and not on word graphs.

2.4 Summary

The basic principles of speech recognition were briefly discussed in the first part this chapter. The process of speech reconition was reviewed from a modular standpoint, and those modules directly related to the contents of this thesis were discussed in more detail.

In the first part of the chapter a generic recognizer was reviewed. In the second part we concentrated on the Daimler-Benz speech recognition system, which was used for all experiments presented in this thesis. As before, those modules directly affected by our work were stressed.

Finally, the evaluation measures needed to understand the experimental results of this thesis were summarized. These measures include word accuracy and word error rate, test set perplexity, out-of-vocabulary rate, and recall and precision.

3 ANALYSIS OF THE OOV-WORD PROBLEM

THIS chapter will analyze the problem of out-of-vocabulary (OOV) words. First, we will show the significance of this problem in speech recognition systems. We will then study some properties of OOV words which will provide us with useful information for detecting them. Finally, we will examine ways of avoiding the occurrence of OOV words.

3.1 Introduction

In this chapter we will study the nature of the OOV-word problem. For this, we will look at various German corpora in order to reach general conclusions which are independent of the particular domain. The corpora include a collection of human-machine dialogs dealing with train timetable inquiries (the EVAR corpus), a collection of human-human spontaneously spoken dialogs in a scheduling domain (the VERBMOBIL corpus), and a collection of written newspaper texts (the TAZ corpus). These corpora are presented in more detail in Section 3.3.

As suggested by its name, out-of-vocabulary words are intimately related to the definition of words, and also to the selection of the vocabulary. In this chapter we will consider a word to be a character string delimited by blanks. As we will show in Chapter 4, this is a simplistic definition, which might not be adequate for characterizing German. But since this is the most widely used definition, as well as the most intuitive one, we opted for it in this study. In Chapter 4 we will present some alternative definitions of words, and will discuss their advantages and disadvantages.

The second aspect closely related with OOV words is the definition of the vocabulary. There are basically three ways of constructing a vocabulary: manually, automatically from the training data, or a combination of both. Usually, for small-vocabulary command-and-control applications of speech technology, the vocabulary is determined by the application and can be constructed by hand. Consider, for instance, a speech recognizer intended for dialing telephone numbers: The numbers, as well as common commands, must be included in the vocabulary. After the system has been tested with some users, the vocabulary can be updated to include the most frequent words uttered by the users which were not originally included.

In research systems, however, vocabularies are usually constructed automatically from the training data. This can be done whenever large databases of transliterated data[1] are available. In this case, all words that appear at least n times in the corpus are included in the vocabulary. (The *vocabulary size* and the *corpus size* are obviously different, since the corpus might contain the same word many times, whereas the vocabulary includes only one entry of each word.) This procedure was used for example in determining vocabularies for the yearly evaluations in the VERBMOBIL project, which contain all words that occurred at least once ($n \geq 1$) in the training corpus defined for that evaluation [Lehning, 1995].

If the amount of training material available is not constant, but grows in time as the data collection evolves, vocabularies consist of the words contained in the training material at a given time. This is a very common situation in large research projects dealing with speech technologies, where the huge amounts of data required for training are not available at the beginning of the project. In this framework, vocabularies are defined in a static manner: One can think of them as a snapshot of the corpus at particular time. In the current study we are interested in the *dynamics* of the data collection process, rather than in a *static* definition of the vocabulary. Understanding this dynamic process will provide interesting information on the nature of OOV words.

We will first present a study of the relationship between corpus size and vocabulary size (Section 3.4.1). There, the corpora will be divided in many small segments, and every such segment will be incorporated into the training corpus sequentially. Thus, the training corpus will grow concurrently with the data collection process. At the same time, the vocabularies computed at every stage of this process will also grow. From these numbers, the vocabulary growth rate will be computed for different corpora, and the differences and similarities among them will be discussed.

Later, we will slightly modify this setup to compute the OOV rate of all incrementally generated vocabularies (Section 3.4.2). Some of the questions we want to answer in this section are: Can the OOV rate be reduced continuously by increasing the vocabulary size? Or is there a saturation point, where a further increase in the vocabulary size does not reduce the OOV rate? And more generally: Can

[1] Throughout this thesis we follow the convention used in the VERBMOBIL project and reserve the term "transcription" for phonetic transcription only. Orthographic transcriptions are denoted as "transliterations."

the OOV-word problem be reduced by lexical means down to a point where it does not affect the performance of the system? We will show that the OOV rate can be substantially reduced by increasing the vocabulary size. But even with very large vocabularies, the OOV rate is too high to ignore.

Having shown the magnitude of the OOV-word problem, we will then investigate some characteristics of OOV words (Section 3.5). In particular, we are interested in a coarse classification of OOV words. If OOV words are mainly semantically important words, their meaning will be of utmost importance for many speech recognition applications, and some mechanism for dealing with them in an adequate way will be indispensable. Our experiments (including those presented in Chapter 4) will show that semantically important words constitute a major portion of the OOV words. We therefore conclude that dealing with OOV words properly is absolutely necessary.

Since OOV words are quantitatively and qualitatively so important for speech recognition applications, we wanted to characterize some of their acoustic properties that could play a role at detecting them. The length distribution of out-of-vocabulary words was investigated and compared with the length distribution of in-vocabulary words (Section 3.5.2). Consistent with previous research done in other languages (see Section 3.2), it was found that OOV words are only slightly longer than frequent words in terms of number of phonemes and number of syllables.

In addition, the magnitude of the acoustic novelty present in OOV words was investigated. For this, the average number of new syllables (i.e. syllables not yet seen in the training corpus) per OOV word was counted. A greater acoustic novelty would make the detection of OOV words easier. Unfortunately though, the acoustic novelty present in OOV is very low: OOV words are made up mostly of known syllables, as will be shown in Section 3.5.3. Furthermore, the average number of new syllables per OOV word decreases with increasing vocabulary size. This is a good indicator of the difficulties we will confront when trying to detect OOV words in Chapter 5.

Finally, since the best possible way of dealing with OOV words is avoiding them in the first place, we dedicate some effort to the subject of lexical optimization. An optimized vocabulary should be as small as possible, while at the same time reducing the expected number of OOV words present in a representative test set. In Section 3.6 we investigate methods aimed at automatically porting vocabularies from one domain to another.

But first let us review previous publications that deal with the subjects we investigate in this chapter, including research related to the OOV-word problem, disfluencies, and vocabulary portability issues.

3.2 Prior Work

3.2.1 Analysis of Out-Of-Vocabulary Words

As pointed out in [Hetherington, 1995a], there are only a few papers that analyze the problem of OOV words. Most surprisingly, they did not appear until a few years after the first publications dealing with OOV-word detection ([Asadi et al., 1990] ,[Asadi et al., 1991], [Kita et al., 1991]). This delay might be due to the fact that the research community was more concerned with simpler tasks, which were manageable with the resources and the technology of that time (such as the Resource Management task [Price et al., 1988]). These tasks were usually fully constrained by a grammar, which either did not allow OOV words at all, or allowed them in particular slots for open word classes. Therefore, a careful analysis of OOV words might not have seemed to be necessary. When more natural speech recognition tasks based on stochastic grammars (which basically allow *any* word sequence) became technologically feasible, the research community showed increasingly more interest on the problem of OOV words.

One of the first efforts to characterize OOV words was presented in [Suhm et al., 1993]. There, a small subset of the Wall Street Journal corpus of American English [Paul and Baker, 1992] was investigated. First, vocabulary growth and vocabulary coverage analyses were performed for different vocabulary sizes. With a vocabulary of approximately 14,000 words, the OOV rate was approximately 4 %. Then, all OOV words (with respect to the 14 K vocabulary) were classified as `names`, `inflections`, `concatenations`, and `other`. Contrary to a widespread belief, the major proportion of OOV words were found not to be proper names but inflected words (46 %). Names made up only roughly one-fourth of all OOV words. Finally, a word length study was performed. The authors found no significant difference between the average number of phonemes in in-vocabulary and out-of-vocabulary words.

In [Hetherington and Zue, 1993], and later in [Hetherington, 1995a], a thorough characterization of OOV words was presented. For vocabulary-growth and OOV-rate studies many different corpora were used (in [Hetherington, 1995a], nine corpora were investigated, including one French and one Italian corpus). It was found that the resulting vocabulary-growth and OOV-rate curves group in either two or three clusters (depending on the interpretation). When grouping the curves into three clusters, the corpora have the following characteristics in common:

- Human-human written communication corpora (e.g. Wall Street Journal [Paul and Baker, 1992]) show the highest vocabulary growth and OOV rate.

- Human-human spoken communication corpora (e.g. Switchboard [Godfrey et al., 1992]) exhibit a medium vocabulary growth and OOV rate.

- Corpora involving human-to-machine interaction (e.g. ATIS [Price, 1990]) show a low vocabulary growth and OOV rate.

Even though OOV rates are clearly domain-dependent, they are consistently high across a wide range of applications. Hetherington reports that a 1 % OOV rate at the word level means that 17 % of all sentences contain at least one OOV word in the Wall Street Journal and ATIS corpora. Since speech recognizers are confronted with sentences, a rate of one-in-six is obviously a much too frequent phenomenon to ignore.

Furthermore, syntactic part-of-speech tags of the Wall Street Journal and ATIS corpora were also investigated. Proper nouns, common nouns, verbs, and adjectives were found to be the most frequent word classes among low frequency words (which are considered to be good candidates for OOV words). This, again, seems to be domain dependent. Whereas proper nouns constitute one-third of all low frequency words in the Wall Street Journal corpus, and are thus the most frequent class, they make up only one-fifth of all OOV words and are less frequent than common nouns and verbs in ATIS. These numbers differ slightly from those published in [Suhm et al., 1993], but since the amount of material and the methodology used for the study are not the same, strict comparisons are not possible.

Finally, a comparison of some phonological properties of high and low frequency words was presented. There, the average number of phonemes and syllables of low frequency words was found to be slightly larger than in high frequency words in both the Wall Street Journal and the ATIS corpora. This finding also contradicts the results presented in [Suhm et al., 1993]. The conclusions in [Hetherington, 1995a] seem more credible, because they were obtained with much more data.

3.2.2 Analysis of Disfluencies

The term "disfluency" describes a group of phenomena frequently observed in spontaneous speech. In spontaneous speech, people tend to plan what they say while they speak [Shriberg, 1994]. Consequently, speaker intentions frequently change after only a part of a sentence has been uttered, causing a disfluency. Among the most common disfluencies are filled pauses, repetitions, and self-repairs [Shriberg, 1996]. We are interested in disfluencies because they are highly correlated with word fragments. And word fragments are by definition OOV words. In fact, word fragments constitute an important source of OOV words, as will be shown later in this chapter. Until recently, disfluencies have been treated as "noise" by the speech recognition research community. Only when more realistic recognition tasks dealing with spontaneously spoken speech were defined, and large training corpora made available, did this subject begin to receive more attention.

The ATIS corpus was the first speech recognition task to be investigated for disfluencies. In [Hirschberg and Nakatani, 1993] it was found that approximately 5 % of the utterances investigated contain repairs. Furthermore, a major proportion (73.3 %) of all repairs end in a word fragment. In this corpus, word fragments have a relative frequency of 0.58 %.

In the Italian train timetable inquiry system presented in [Baggia et al., 1994], speech disfluencies were also observed in around 5 % of all user utterances. Half of them contained word fragments at the interruption site. This translates into an OOV rate due to word fragments of 0.47 %. The vocabulary of the system contained 762 words, and the overall OOV rate was 2.5 %. Hence, in this task about one-fifth of all OOV words were word fragments.

In [Shriberg, 1994], a thorough study of speech disfluencies was presented. There, three corpora were investigated: ATIS [Price, 1990], AMEX [Kowtko and Price, 1989], and Switchboard [Godfrey et al., 1992]. The sentence disfluency rate of the ATIS corpus was found to be similar to the figures presented by other researchers (approximately 6 %). The AMEX and Switchboard corpora are much more disfluent: Around one-third of all sentences exhibit disfluencies. Shriberg attributes this to the different scenarios. Whereas in the ATIS corpus human beings interact with computers, both AMEX and Switchboard register conversations between humans. Word fragments were also investigated. Surprisingly, ATIS has the highest rate of disfluencies with fragments (roughly 60 %). In AMEX and Switchboard only about one fourth of all disfluencies have a word fragment. Again, this seems to be related to the fact that ATIS involves human-computer interaction.

Schultz and Rogina reported their experiences when adapting a speech recognizer to the particularities of spontaneous speech [Schultz and Rogina, 1995]. Experiments were conducted on a rather small set of dialogs of the English Spontaneous Scheduling Task, the English counterpart of VERBMOBIL. The authors report that word fragments constitute 0.69 % of the corpus. The overall OOV rate of the test set was not reported.

3.2.3 Vocabulary Portability

Hetherington not only presented an excellent analysis of the OOV-word problem, but also studied the possibility of porting vocabularies from one domain to another [Hetherington and Zue, 1993], [Hetherington, 1995a]. He used a straightforward maximum-likelihood approach to select vocabularies: All words having a frequency f greater than a given threshold n in the training material were included in the vocabulary.

First, the Wall Street Journal corpus (WSJ) was used for constructing vocabularies and ATIS for testing coverage [Hetherington and Zue, 1993]. As a reference, the domain-specific ATIS coverage was also reported (that is, the coverage of a vocabulary obtained from ATIS transliterations). With a similar vocabulary size, the WSJ vocabulary showed a OOV rate approximately 60 times higher than the reference ATIS vocabulary. WSJ vocabularies had to be roughly one order of magnitude larger than the domain-specific ATIS vocabulary to achieve a similar coverage. Enlarging the WSJ vocabularies further resulted in a lower OOV rate than with the domain-specific ATIS vocabulary, which was used as a reference.

In his PhD thesis, Hetherington studied vocabulary portability issues of much more similar corpora: the Wall Street Journal, and the New York Times newspapers [Hetherington, 1995a]. Cross-task OOV rates were found to be approximately three times higher than task-specific OOV rates. Thus, the author concluded that porting a vocabulary from one task to another similar task can result in a significant increase in the OOV rate. These results appear to be rather pessimistic at first glance, especially compared to the results obtained in the much more difficult scenario presented in [Hetherington and Zue, 1993] and mentioned above.

One possible explanation for this outcome is the problem of *diminishing returns*, which is stated as follows: "Very low frequency words in one task are not very useful in reducing the OOV rate in another task. Presumably, these low frequency words are more task dependent than higher frequency words" [Hetherington, 1995a, page 49]. When measuring the coverage of the small-vocabulary ATIS test set, mainly high frequency words from the WSJ corpus were used. By contrast, when comparing the WSJ and New York Times corpora, many low frequency words are included in the vocabularies, making them less portable.

Rosenfeld studied the process of creating a vocabulary of a given size, which minimizes the OOV rate of the test data [Rosenfeld, 1995c]. He did this by selecting the training material using different criteria, and generating a word list ordered by decreasing frequency of occurrence in the selected training material. From this ordered word list, vocabularies of different sizes were extracted, and their coverages of the test set were subsequently tested.

The North-American-Business corpus [Rosenfeld, 1995a] was used for training purposes, and coverage measurements were performed on a subset of it. The author found no significant effect of the seasonality (i.e. the time of the year) the training material is from. The amount of training material and its recency do have a positive influence on the coverage of the resulting vocabularies. Combining the effect of these two factors, the author achieved the best coverage of the test set by using only about one fifth of the training material available, namely the most recent part. Also the influence of the source of the training material was investigated. Consistent with the research work discussed in the previous paragraphs, it was found that the highest coverage is achieved when the training and test vocabularies are extracted from the same newspaper.

3.3 The Corpora

Three corpora that will be used in the analysis presented in this chapter, as well as elsewhere in this thesis will be introduced. We deal with corpora in its textual, orthographic form. Even though acoustic information is available (for the EVAR and VERBMOBIL corpora), it will not be used for the purposes of this chapter. The language of all three corpora is German.

3.3.1 The EVAR Corpus

The EVAR corpus was recorded at the University of Erlangen, Germany, as part of an effort to provide a natural language interface to a train timetable database [Kuhn, 1995],[Eckert et al., 1995]. First, a set of automatically generated sentences was recorded to bootstrap the acoustic models of the recognizer. Afterwards, a first version of the system was made available over the public telephone network, and all interactions with users were registered. After the recordings were transliterated[2], this material was used to update the system models. This is also the portion of the corpus we used for our analysis.

A typical session with the system is shown in Figure 3.1. As can be seen, the dialog structure is relatively simple, and fairly constrained to the application domain (although this is not the case with some uncooperative users). Table 3.1 summarizes the magnitudes of the subset of the EVAR corpus made available to us.

Corpus	Turns	Words	Vocabulary
EVAR	10,556	$\approx 38\,\mathrm{K}$	$\approx 1.3\,\mathrm{K}$

Table 3.1: Some general statistics of the EVAR corpus.

S: Guten Tag. Hier ist die automatische Intercityauskunft in Ulm. Wohin möchten Sie fahren?	S: Hello. This is the train information system in Ulm. Where do you want to go to?
U: *ich möcht' morgen zum Frankfurt Flughafen fahren.*	U: *I wanna go to Frankfurt airport tomorrow.*
S: Sie möchten von Ulm nach Frankfurt Flughafen fahren. Wann wollen Sie abfahren?	S: You want to go from Ulm to Frankfurt Airport tomorrow. When do you want to depart?
U: *ich muß vor'm Mittag da sein.*	U: *I've gotta be there before noon.*
S: Der ICE 123 fährt von Ulm Hauptbahnhof um 9:13 ab und kommt in Frankfurt Flughafen um 11:34 an. Benötigen Sie weitere Informationen?	S: You can take the ICE 123 at Ulm Central Station at 9:13. you will arrive at Frankfurt Airport at 11:34. Do you need any more information?
U: *nein, danke. wiederhören.*	U: *no thanks. bye.*

Figure 3.1: An example of a typical dialog in the EVAR scenario. The real German dialogs are shown on the left side. A rough English translation is given on the right. The user utterances (labeled **U:**) are printed in italics.

3.3.2 The Verbmobil Corpus

VERBMOBIL is a simultaneous speech-to-speech translation system aimed at supporting human-human communication [Wahlster, 1993], [Bub and Schwinn, 1996], [Bub et al., 1997]. In the VERBMOBIL scenario, a German and a Japanese person discuss the date of an appointment in English, and use the VERBMOBIL device as a translation aid.

For training the German speech-understanding modules, many spontaneous speech dialogs were recorded at various universities throughout Germany. The data was collected over the entire duration of the project, and was delivered to the project partners at regular intervals. The VERBMOBIL corpus grew continuously from less than a hundred dialogs to over four hundred dialogs by the end of the first project phase. At the time this thesis is being written, the second VERBMOBIL phase is starting. In this phase, further recordings with significant scenario variations with respect to the first phase are planned. Our work, though, restricts itself to the VERBMOBIL corpus collected during the first project phase.

VERBMOBIL is a corpus of spontaneously spoken speech in human-human dialogs. Therefore, it contains a series of phenomena not found in read, planned

[2] As mentioned above, orthographic transcriptions are termed “transliterations” throughout this thesis.

WEG: ja, guten Tag, Frau Pätzold, <Pause> Schneidereit hier. <Schmatzen> <Atmen> <äh> wir hatten<Zögern> uns noch <ähm> wegen des<Zögern> Kurses Wirtschaftsenglisch mal zusammensetzen wollen. wollen Sie zu mir rüberkommen oder sollen wir das jetzt am Telefon machen <#Klicken>?	WEG: hello Ms Pätzold, <Pause> Schneidereit speaking. <Smack> <Breath> <uh> we wanted to<Linger> get together <hm> on this<Linger> business English class. do you want to come over to my place or should we take care of it now on the phone <#Click>?
ANP: <Atmen> <Schmatzen> oh, <:<#Clicken> können:> wir ruhig jetzt hier besprechen, also, ich hab' jetzt auch nicht so viel Zeit <#Klicken>.	ANP: <Breath> <Smack> oh, <:<#Click> we could:> do it right now. well, I really don't have a lot of time right now <#Click>.
WEG: <Atmen> <Schmatzen> okay. <Atmen> <ähm> wir haben ja acht Termine<Zögern>, die wir noch verteilen müssen<Zögern>, und<Zögern> <ähm> da erstmal +/eine/+ meine Frage zuerst, <Atmen> machen wir %das en <:<Atmen> bloc<Zögern>:> als<Zögern> <ähm> festen Tag in der Woche<Zögern> oder<Zögern> <äh> mehrere Einzelblöcke oder wie haben Sie sich das gedacht <#Klicken>?	WEG: <Breath> <Smack> okay. <Breath> <uhm> we have eight appointments<Linger> we have to fit in somewhere<Linger>, and<Linger> <uhm> since first +/we/+ first my question, <Breath> should we do %it all <:<Breath> together<Linger>:> on a fixed day of the week<Linger> or<Linger> <ah> several single blocks or what do you think <#Click>?

Figure 3.2: An example of a section of a typical dialog in the VERBMOBIL scenario. The real German dialogs are shown on the left side. A rough English translation is given on the right.

speech. Among these phenomena are hesitations and filled pauses, human and non-human noises, simultaneity of acoustic events, unintelligible segments, pronunciation variants, disfluencies, word fragments, and repairs. These phenomena were systematically captured in the transliterations provided with the recordings in order to facilitate research in spontaneous speech with a large-scale corpus [Kohler et al., 1994]. Figure 3.2 shows a section of a typical dialog in the VERBMOBIL scenario.

The tags used for spontaneous speech events in the VERBMOBIL corpus are described in Table 3.2. These tags were filtered appropriately to facilitate training acoustic and language models. Slightly different conversions were applied to generate the reference transliterations for evaluating the accuracy of the recognizer.

Event	Tag and example	Mapping
Silence	<Pause>	—
Breath	<Atmen>	—
Smack	<Schmatzen>	—
Linger	drei<Zögern>	drei
Hesitation	<äh>	—
Non-verbal articulatory production	<Lachen>	—
Non-articulatory noise	<#Klicken>	—
Simultaneity of acoustic events	:<#Klicken> können:>	können
Pronunciation variant	nicht <!1 net>	nicht
Contracted word	'n bißchen	ein bißchen
Split word	Arbeits_<Pause>_besuch	Arbeitsbesuch
Repair	+/eine/+	eine
Word fragment	Mitt=	<UNK>
Technical interruption	aus<;T>	<UNK>
Non-word or neologism	*roger	<UNK>
Difficult to understand	%das	das
Non comprehensible	<%>	—

Table 3.2: Some of the tags used to mark spontaneous speech events in the Verbmobil corpus. In the third column, the conversions applied to obtain the reference transliterations are shown.

Table 3.2 shows the mappings applied to obtain the reference transliterations. Table 3.3 shows some general statistics of the reference transliterations in the Verbmobil corpus.

Corpus	Dialogs	Turns	Words	Vocabulary
Verbmobil	789	13,892	$\approx$ 314 K	$\approx$ 6 K

Table 3.3: Some general statistics of the Verbmobil corpus.

3.3.3 The TAZ Corpus

The ***Ta**geszeitung*, short TAZ, is a German newspaper published in Berlin, Germany. An electronic version of everything published from July 1, 1988 to June 30, 1994 is available from the publishers (for more information, see http://www.taz.de). At the time this study began, it was the largest collection of German texts in electronic form.

Since the database was provided with formatting information, it had to first be converted into an appropriate format for lexical and language model research. General statistics of the resulting text collection are given in Table 3.4. There, the vocabulary size may seem strikingly large. We first thought that this was caused by a large number of spelling errors in the original texts. By inspecting randomly selected samples of the approximately 0.7 million singletons[3] in the corpus, we confirmed that indeed some spelling errors are present, but they do not make up a major portion of all singletons. Next, we suspected that this is partially due to word compounding and inflections, which are very frequent in German. This moved us to continue investigating morphological aspects of the corpus. A detailed study of these aspects will be presented in Chapter 4.

Corpus	Sentences	Words	Vocabulary
TAZ	$\approx 3.9\,\mathrm{M}$	$\approx 76\,\mathrm{M}$	$\approx 1.3\,\mathrm{M}$

Table 3.4: Some general statistics of the TAZ corpus.

3.3.4 Summary

We have presented three corpora that will be used for the analyses discussed in the current chapter. The corpora include a collection of human-machine dialogs dealing with train timetable inquiries (EVAR), a collection of human-human spontaneously spoken dialogs in a scheduling domain (VERBMOBIL), and a collection of written newspaper texts (TAZ). Table 3.5 summarizes the most important characteristics of these corpora. Table 3.6 shows the corpus sizes and the vocabulary sizes of the corpora. The TAZ corpus and especially the VERBMOBIL corpus will also be used in subsequent chapters.

3.4 Vocabulary Growth and Coverage

We will now study two characteristics of the corpora presented above. The first is the growth of the vocabulary as a function of the corpus size and domain. This will be presented in Section 3.4.1. Section 3.4.2 will look at how differently sized vocabularies cover unseen data.

[3]Singletons are words that occur only once in a corpus.

EVAR	VERBMOBIL	TAZ
spoken language	spoken language	written language
human-machine	human-human	human-human
spontaneous speech	spontaneous speech	written language
limited domain	limited domain	unlimited domain

Table 3.5: Summary of the main characteristics of the EVAR, VERBMOBIL, and TAZ corpora.

Corpus	Corpus size	Vocabulary size
EVAR	38 K	1.3 K
VERBMOBIL	314 K	6 K
TAZ	76 M	1.3 M

Table 3.6: Some statistics of the EVAR, VERBMOBIL and TAZ corpora.

3.4.1 Vocabulary Growth

In this section we investigate the relationship between corpus size and vocabulary size. We do this using the three corpora described in detail in Section 3.3. To simulate the growth of each corpus, they are first divided into small segments (dialog turns in EVAR and VERBMOBIL, and sentences in TAZ). The order of the segments is then randomized and processed in sequential steps. In every step, an additional segment is incorporated into the training corpus, and the total number of words in the training corpus as well as the size of its vocabulary is registered. This process continues until all segments are incorporated into the training corpus. Finally, the whole process is repeated many times. Our method is very similar to the one used by Hetherington [Hetherington, 1995a][4]. This was done intentionally in order to allow direct comparisons with his results.

The results of our experiments are shown in Figure 3.3. The different characteristics of the three corpora studied result in clearly differenciated vocabulary growth rates:

- The TAZ corpus, intended for human-human written communication, has the highest vocabulary growth rate.

[4] The main difference is that Hetherington *averaged* the results obtained in each experiment iteration. We believe that averaging hides the fact that we are dealing with random variables, whose values can only be predicted with a certain probability to be within a confidence interval. Therefore, we do not average experiment iterations, but plot them all individually in our graphs.

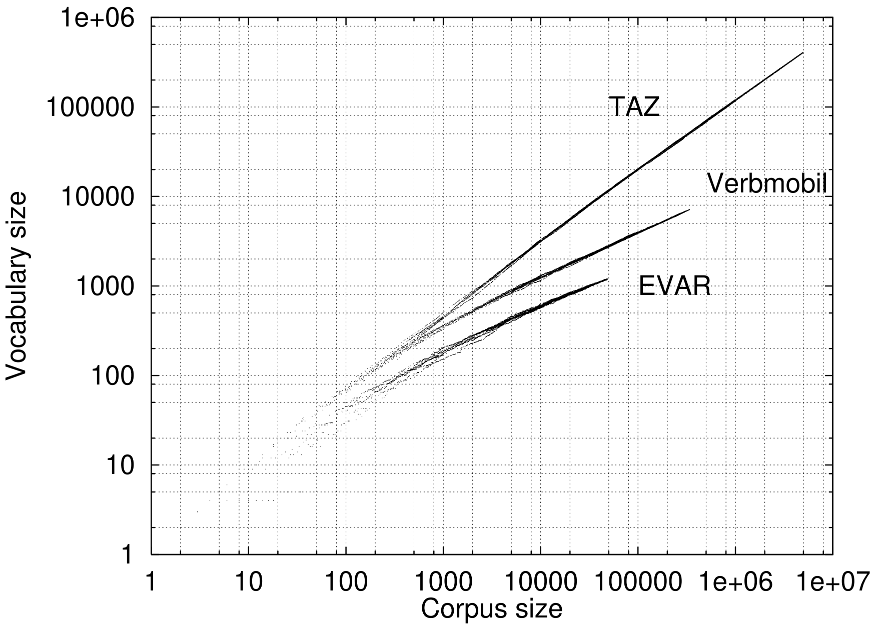


Figure 3.3: Vocabulary growth of the TAZ, VERBMOBIL, and EVAR corpora. For better visualization, only a part of the TAZ corpus is shown.

- The VERBMOBIL corpus, consisting of human-human speech dialogs in a constrained domain, exhibits a medium vocabulary growth rate.

- The EVAR corpus, which registers the interaction of human beings with computers, has the lowest vocabulary growth rate.

This outcome is consistent with Hetherington's findings [Hetherington, 1995a]. As mentioned in Section 3.2.1, he found that the vocabulary growth depends on the application domain, but is basically independent of the language, at least for English, French, and Italian. We presume that the German language will have much larger vocabulary growth rates due to its high morphological flexibility. This fact has already been discussed in other publications (see, for instance [Geutner, 1995]), but was not yet investigated using corpora of a considerable size.

In order to examine this, we chose corpora with a similar domain but different language: the German TAZ corpus and the American New York Times (NYT) and Wall Street Journal (WSJ) corpora[5]. If Hetherington's conclusion on the language independence of the vocabulary growth rate is also valid for German, we would

[5] A comparison with the VERBMOBIL or EVAR corpora is not possible, since no similar domains are reported in [Hetherington, 1995a].

expect these corpora to exhibit roughly the same rates. The TAZ corpus, however, has much larger vocabulary sizes than either NYT or WSJ for a given corpus size. For example, in order to create a vocabulary of 100,000 words one needs 10 million WSJ words, but only 1 million TAZ words of training material. This motivated us to further investigate subjects related with German morphology, which will be presented in Chapter 4.

3.4.2 Coverage and Out-Of-Vocabulary Rate

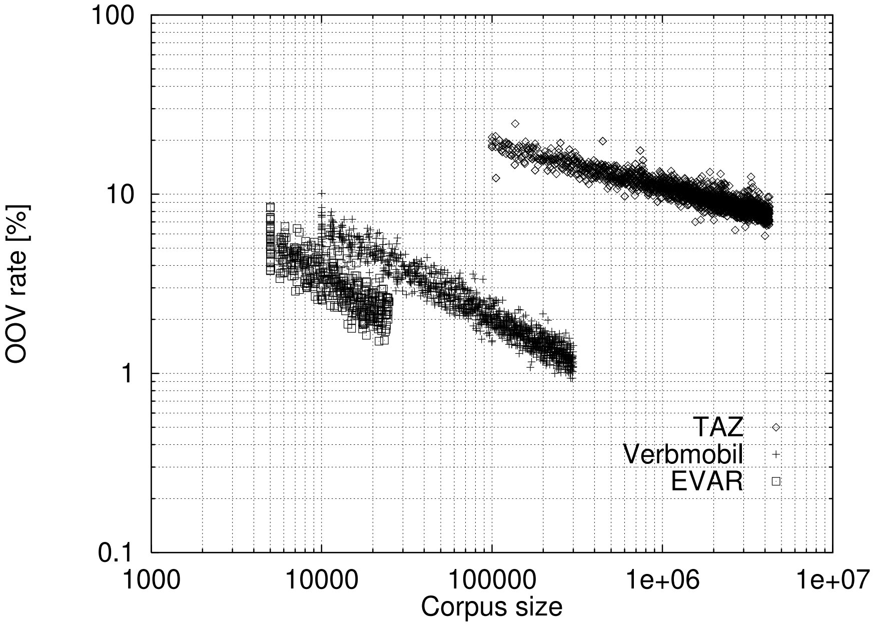
In the previous section we described the relationship between the size of a corpus and the size of the vocabulary contained in the corpus. So for a given corpus size we can estimate the size of the vocabulary that can be automatically obtained from the corpus. The question is: How efficiently will this vocabulary cover unseen data? In other words, how many OOV words are expected in a random test set of a given size?

In order to investigate this, the procedure described in the last section was slightly changed. As before, we first randomized the order of the turns (or sentences, for TAZ). Then, instead of turns (or sentences), we used larger text portions as segments: The segments were enlarged by incorporating more turns (or sentences) until the number of OOV words within the segment was larger than a given threshold. Then, the segments are processed sequentially. In every step, we registered all words which had not occurred previously as out-of-vocabulary and then add them to the vocabulary, thus making them *in-vocabulary* for subsequent segments. This process continues until all segments are incorporated into the training corpus. Finally, the whole process is repeated many times, and all outcomes are plotted.

The top of Figure 3.4 shows the OOV rate as a function of the corpus size. The bottom of Figure 3.4 shows the OOV rate as a function of the vocabulary size. As expected, corpora with a high vocabulary growth rate, also have high OOV rates. This is the case with the TAZ corpus. The VERBMOBIL and the EVAR corpora have much lower OOV rates. From Figure 3.4, it is clear that the expected OOV rate of unseen data can be significantly reduced by increasing the vocabulary size. But it is also clear that a complete coverage cannot be obtained.

If the test set used to measure the OOV rate is *not* chosen randomly, the OOV rates might be different from the numbers shown in Figure 3.4. For instance, if the test set of a spoken corpus includes only new speakers not yet seen in the training corpus, we expect the OOV rates to be higher. Experimental results confirmed this conjecture (see Chapter 5).

The rate of turns or sentences containing OOV words is also of interest. This measure is relevant because speech recognition systems are confronted with whole



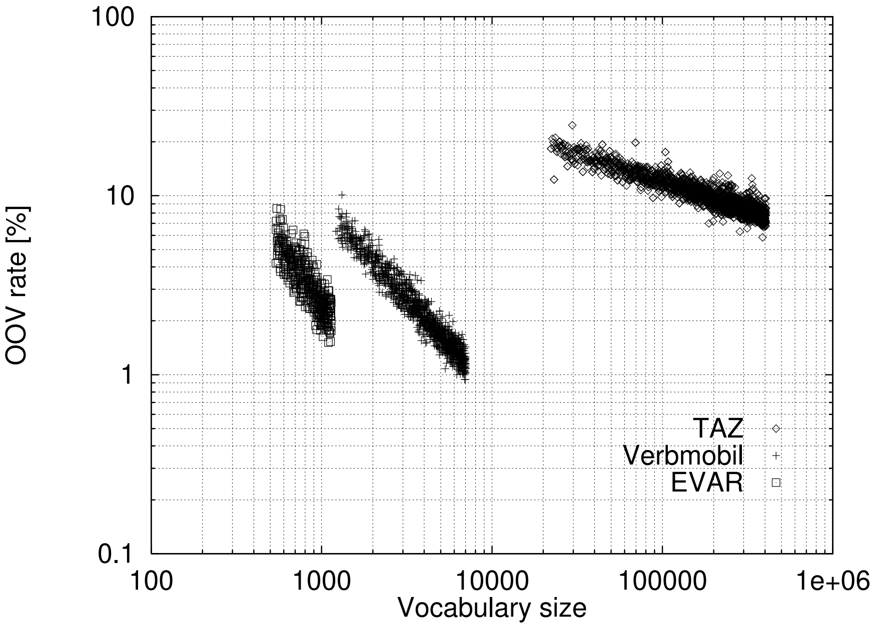


Figure 3.4: OOV rate as a function of the corpus size (top), and as a function of the vocabulary size (bottom). The TAZ, VERBMOBIL, and EVAR corpora are depicted in both plots.

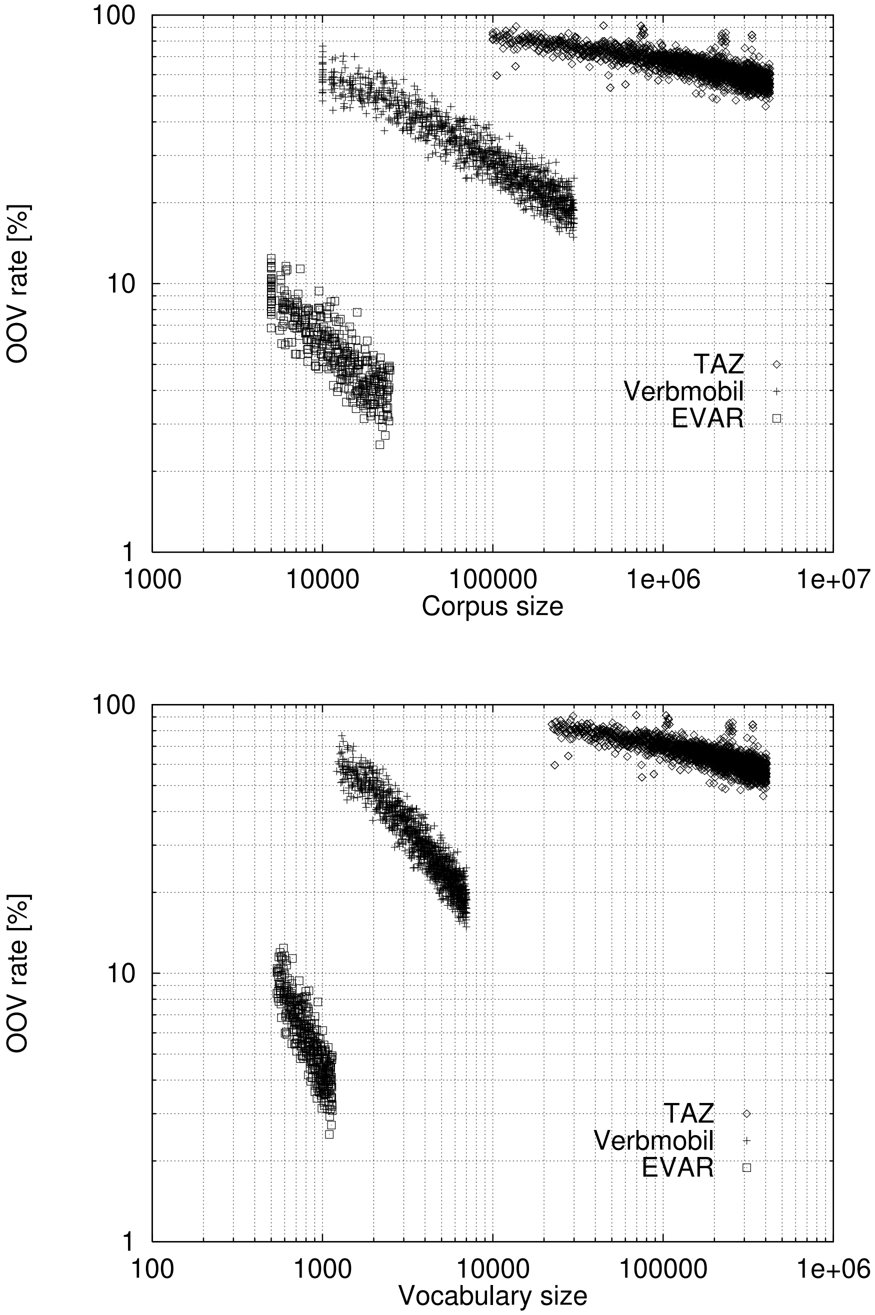


Figure 3.5: OOV rate per turn (for the VERBMOBIL and EVAR corpora), and OOV rate per sentence (for the TAZ corpus) plotted as a function of the corpus size (top), and vocabulary size (bottom).

turns or sentences, not with isolated words. For this analysis, the VERBMOBIL and EVAR corpora were separated in turns, and the TAZ corpus in sentences.

Figure 3.5 shows the results. Since the average number of words per turn (or sentence) varies widely among the corpora[6], the plots in Figure 3.5 differ from those in Figure 3.4. In the VERBMOBIL corpus, for example, the OOV rate at the turn level is currently approximately 20 %. In other words, every fifth turn contains on the average one or more OOV words. This is clearly much too frequent to ignore.

Cutoff frequency

A further interesting aspect of vocabularies created automatically from training data is the threshold frequency used to incorporate a word into the vocabulary. This frequency is usually called *cutoff frequency* f_{cutoff}. A word w is included in the vocabulary if its frequency is greater than the cutoff frequency $f_w > f_{\text{cutoff}}$. In all experiments performed so far, all words in the training corpus were incorporated in the vocabulary. In other words, a cutoff frequency $f_{\text{cutoff}} = 0$ was used.

In order to investigate this, we repeated our previous experiments using various cutoff frequencies. The results obtained for the VERBMOBIL corpus[7] are depicted in Figure 3.6. The experiment setting used to generate the scatter plot labeled Cutoff = 0 is the same as that used in the VERBMOBIL plot of Figure 3.4 (although the plotting parameters are slightly different). The plot labeled Cutoff = 9 includes all words that occur at least 10 times in the training corpus. This figure clearly shows that frequency information can be very beneficial for reducing the OOV rate: For a given vocabulary size (say 1,000 words) the OOV rate of the vocabularies that include only frequent words (Cutoff = 9) is significantly lower than the vocabularies that include all words seen so far (approximately 5 % and 8 % respectively). But, of course, in the former case much more training material is required for constructing a vocabulary.

Incremental and Batch Vocabularies

Finally, we also investigated the difference between *incremental* and *batch* vocabulary construction. Incremental vocabulary construction is what has been described so far: We consider only a portion of the corpus to be known, and use it to create a vocabulary. A batch vocabulary, on the other hand, is constructed all at once using the whole corpus.

[6] Whereas EVAR turns have on the average less than 4 words (see Table 3.1), VERBMOBIL turns have over 20 (see Table 3.3).

[7] The results of similar experiments with EVAR and TAZ are not shown in Figure 3.6 for clarity. The conclusions arrived for the VERBMOBIL corpus are also valid for EVAR and TAZ.

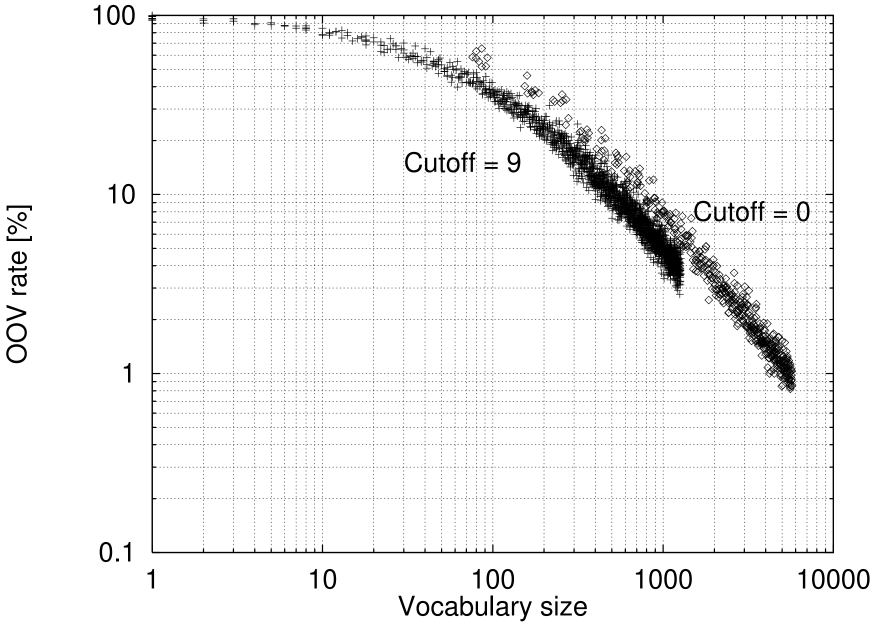


Figure 3.6: Influence of cutoff frequencies on the OOV rate of the VERBMOBIL corpus. The plot labeled Cutoff = 9 only includes words that occur at least 10 times in the training corpus. The plot labeled Cutoff = 0 includes all words in the corpus.

In order to compare incrementally generated vocabularies with "incrementally growing" batch vocabularies, the following procedure is applied. First, the batch vocabulary is sorted in descending word frequency. The first batch vocabulary consists of the first word (that is, the most frequent word), as well as all following words with the same frequency. For each succeeding vocabulary, the next word in the sorted list is added along with all following words with the same frequency. In effect, every possible cutoff frequency is used to create a vocabulary. In every step, the OOV rate of the test set was computed, and, as before, this process was repeated for different test sets. Since in this case we intended to evaluate the highest coverage obtainable for a given vocabulary size, we used closed testing conditions, that is, the training set included the test set.

Figure 3.7 shows the results, which are consistent with the outcome of the previous experiment (Figure 3.6). Here, again, the batch vocabularies cover the test set much better. This confirms our conclusion that using word frequency information when building vocabularies can significantly reduce the OOV rate.

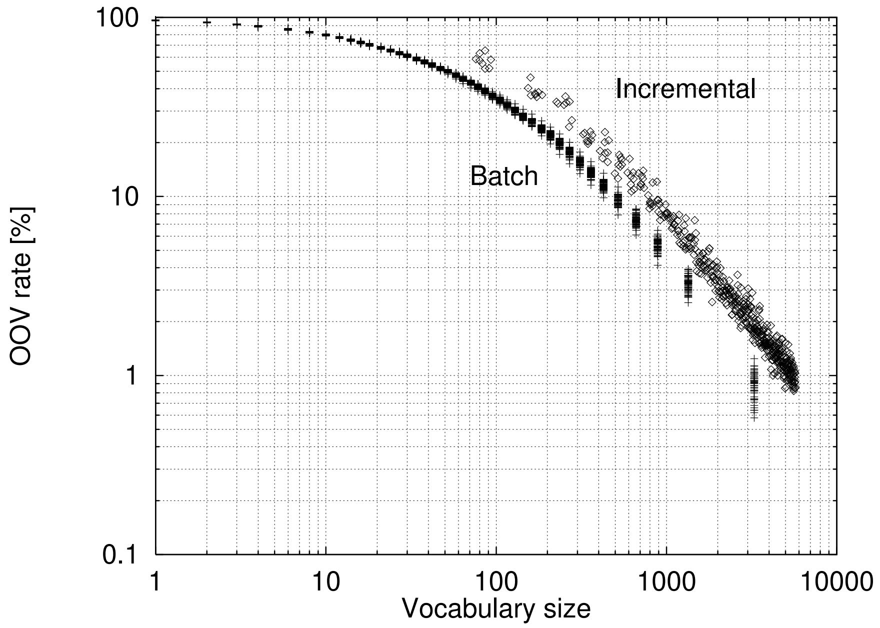


Figure 3.7: Incremental and batch vocabulary construction in the VERBMOBIL corpus. Incremental vocabularies include only words so far seen in the training corpus. Batch vocabularies include all words that occur more than n times in the entire corpus.

3.5 Properties of Out-Of-Vocabulary Words

The last section showed that the occurrence of OOV words is inevitable. In this section we want to gain more insight to the problem of OOV words. Therefore, a coarse functional classification and a characterization of acoustic properties of OOV words will be now presented. For this analysis we will restrict ourselves to the VERBMOBIL corpus, which will be used throughout this thesis for recognition experiments.

3.5.1 OOV-Word Classification

Having shown the quantitative importance of OOV words, we are now interested in a qualitative characterization. For this, all words in the VERBMOBIL vocabulary were classified into three classes:

Proper names: include names of persons, cities and places. The words belonging to this category were tagged by hand in the VERBMOBIL vocabulary.

Word fragments: include not only the type of fragments related to disfluencies, but also so-called *technical* interruptions, caused when recording was initiated too late or interrupted [Kohler et al., 1994]. This word class was tagged in the VERBMOBIL transliterations (see Table 3.2).

Other words.

This classification is especially suited to the VERBMOBIL scenario. In VERBMOBIL, proper names play a central role and word fragments require special treatment. In Chapter 4, we will use morphological knowledge to classify the VERBMOBIL vocabulary still further.

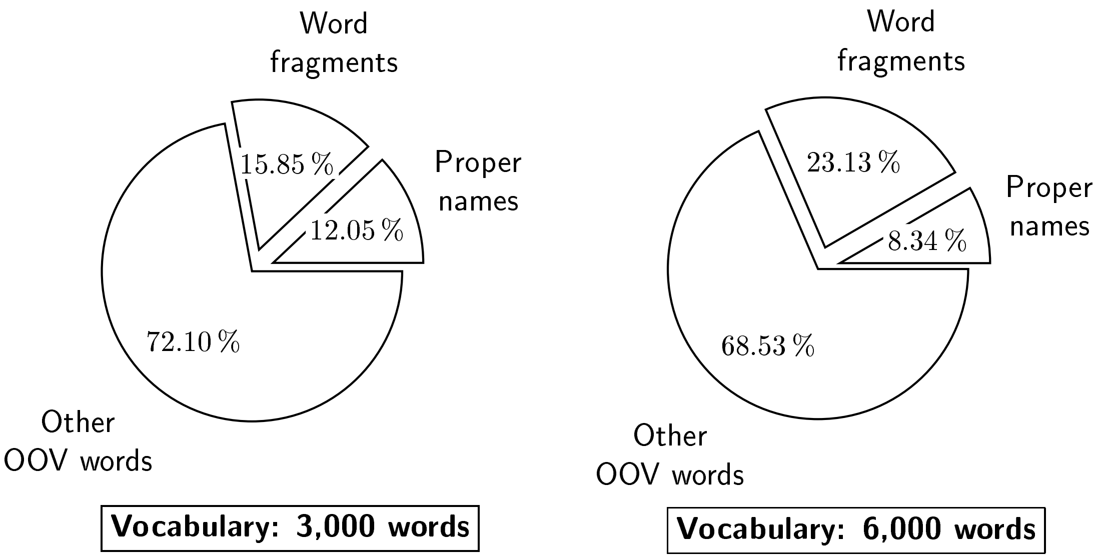


Figure 3.8: Classification of OOV words in Proper names, Word fragments, and Other OOV words for vocabularies of 3,000 (left) and 6,000 words (right).

In order to simulate the occurrence of OOV words, we used the procedure presented in Section 3.4.2. Here, the contribution of each class to the total OOV rate was also calculated. Then, we computed a linear regression over all experiment iterations, and drew pie charts for vocabularies of 3,000 and 6,000 words (see Figure 3.8).

As can be seen, the size of the vocabulary has only a small influence on the significance of each class. Surprisingly, proper names make up only a minor portion of all OOV words (around 10 %). Roughly one fifth of the OOV words are word fragments, a figure consistent with other spontaneous speech tasks reported in the literature (see Section 3.2.2). The overwhelming majority of the OOV words belong to the class "Other OOV words." A quick inspection in this class revealed that many OOV words are affected by some morphological mechanism, such as inflection or compounding. This motivated us to further investigate the effect of morphology on OOV words, and to dedicate a separate chapter to this subject (see Chapter 4).

3.5.2 OOV-Word Length

Since OOV words inevitably occur, we are interested in studying some of their acoustic characteristics. Information on the average length of OOV words could, for instance, be utilized when choosing acoustic models for OOV detection (see Chapter 5). In this section we compare the average length of the vocabulary with the average lentgh of OOV words. The length of words may be measured in terms of the number of phonemes and the number of syllables.

OOV words from the Verbmobil corpus were simulated as described above, and their length in syllables and phones were determined. The measurements performed are shown in Figure 3.9. As reported in previous studies [Hetherington, 1995a], we also found out-of-vocabulary words to be slightly longer than in-vocabulary words. The average difference is approximately one phoneme or one-third of a syllable. Furthermore, the average word length grows continuously as the corpus grows, whereas the length difference between OOV and in-vocabulary words remains roughly constant.

3.5.3 Acoustic Novelty

A further interesting aspect of OOV words is their degree of acoustic novelty. Acoustic novelty directly influences two of the main issues studied in this thesis, namely OOV-word detection and transcription. If OOV words are mostly made up of known acoustic units, discriminating them from in-vocabulary words (i.e. detecting them) will be all the more difficult[8]. On the other hand, since acoustic units which have already been observed in the training set are more likely to be

[8] This assertion does not take the influence of the lexicon into account. It is therefore somehow simplistic. But since it is intuitive and was repeatedly confirmed by recognition experiments, we use it as an indicator.

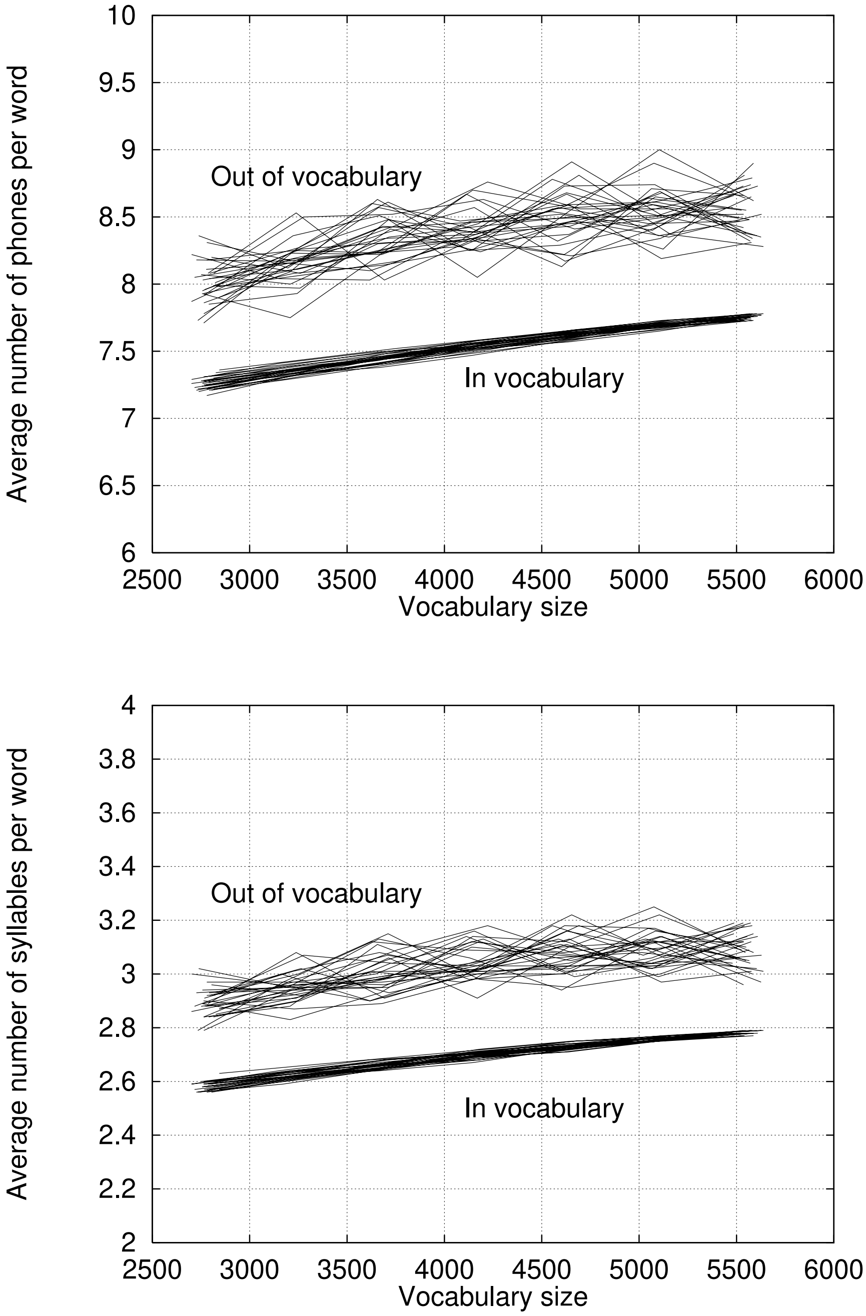


Figure 3.9: Average length of out-of-vocabulary words and in-vocabulary words. The measurements refer to the *vocabulary* of the VERBMOBIL corpus (i.e., word frequencies are not considered when computing the average word length). Word lengths are measured in phones (top) and syllables (bottom).

well trained, transcribing OOV words (and thus incorporating them in the system lexicon[9], as will be discussed in Chapter 6) will be simpler.

To simulate OOV words we used the same algorithm as in previous sections. The average number of new syllables per OOV word was chosen as indicator of acoustic novelty. When a simulated OOV word is encountered, it is decomposed in its syllables, and each of them is looked up in the list of syllables observed so far. If the syllable in question is not found, it is registered as unknown, and incorporated into the list of known syllables at the end of the current iteration. This process is then repeated several times.

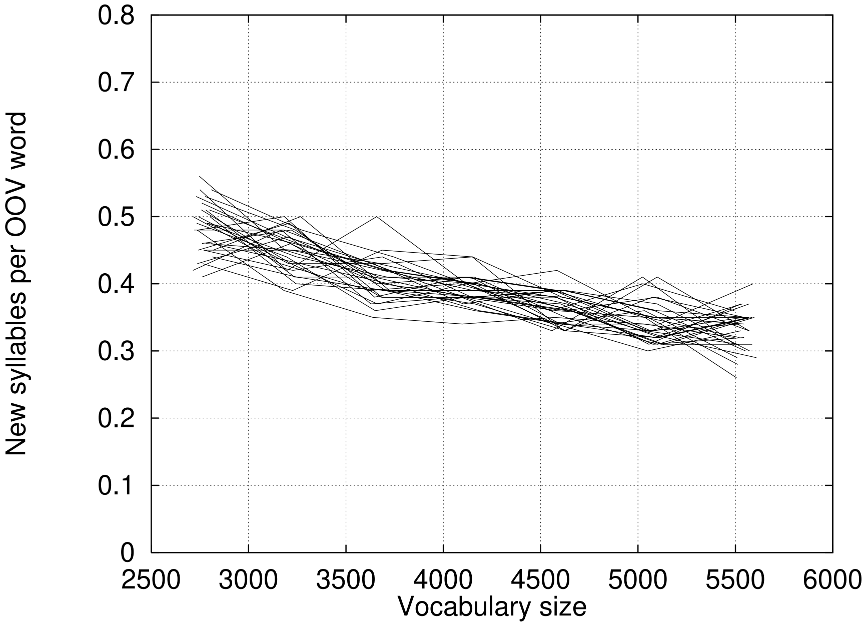


Figure 3.10: Acoustic novelty in OOV words measured in the average number of new syllables per OOV word.

From the results in Figure 3.10 it can be seen that the acoustic novelty present in OOV words is very low: On average, between a half and a third of a syllable per OOV word is new. Furthermore, the number of new syllables per OOV word decreases with increasing vocabulary sizes. Combining this result with those shown in Figure 3.9, we see that only 10 % to 20 % of the acoustic sequences (measured in syllables) contained in OOV words are new. This is a good indicator of the

[9]The terms *vocabulary* and *lexicon* are sometimes used as synonyms. Strictly speaking, though, a vocabulary is merely a list of words, whereas a lexicon contains additional information about each word (its phonetic transcription, for example), as discussed in Chapter 2.

difficulties involved in trying to detect OOV words (see Chapter 5).

3.6 Vocabulary Portability

As was shown in Section 3.4, OOV words appear in a wide range of domains. Moreover, even though the OOV-word problem cannot be completely solved by increasing the vocabulary size, a significant improvement in the vocabulary coverage can be achieved this way.

The importance of avoiding OOV words in speech recognition was shown, for example, in the 1994 CSR Hub Evaluation of ARPA. In this evaluation, the test conditions were given by the North-American-Business corpus (which includes the Wall Street Journal corpus), and the lexicon size could be freely chosen by the participants. As discussed in [Rosenfeld, 1995b], the biggest improvement in word error rate in almost all participating sites came from increasing the recognition vocabulary from 20,000 to approximately 64,000 words. This was possible in the North-American-Business task because there was sufficient domain-specific material available.

A very common problem in speech recognition tasks dealing with spontaneous speech, though, is the lack of domain-specific training material. The data collection process is usually slow and expensive, since recordings have to be manually transliterated and corrected. Therefore, lexica constructed automatically from the available training material tend to be incomplete and exhibit a poor coverage when presented with unseen test material. This can be clearly seen in Figures 3.4 and 3.5.

One apparently obvious way to alleviate this problem is to complement small vocabularies with words obtained from electronically-available dictionaries. However, since standard dictionaries do not contain frequency information, which is domain-dependent, and are also too large for most speech recognition applications, they are not suited to vocabulary optimization [Hetherington, 1995a].

Another possibility is to use electronically available text collections, which do contain frequency information. Unlike the situation described above for speech recognition applications, large text corpora are available for many languages. The questions that arise are: Can information be extracted from these huge text sources that is useful for a particular application domain? Is the difference between domains decisive, or are there domain-independent characteristics that can be used to enhance small corpora?

3.6.1 Porting Vocabularies

In order to investigate these questions, we experimented with the TAZ newspaper corpus, and the VERBMOBIL spontaneous speech data base. We first measured the batch and incremental vocabulary coverage of VERBMOBIL, as introduced in Section 3.4.2. Then, we computed the OOV rate of TAZ vocabularies on VERBMOBIL. TAZ vocabularies were constructed according to the batch procedure described above, i.e., a vocabulary of the size n contains the n most frequent words in the TAZ corpus.

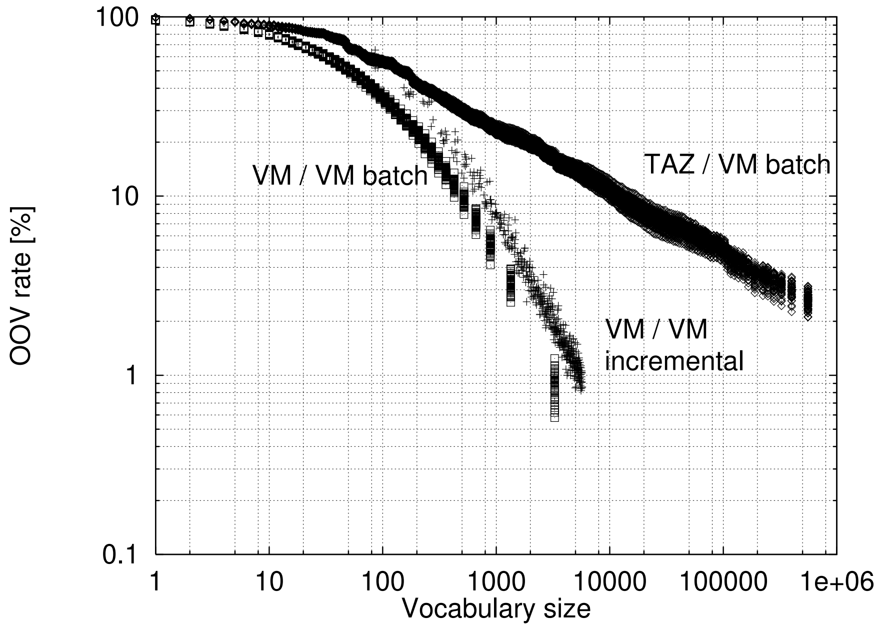


Figure 3.11: Intra-domain versus cross-domain vocabulary coverage. The plots labeled VM/VM show the coverage of both batch and incremental VERBMOBIL vocabularies. The plot labeled TAZ/VM depicts the coverage of TAZ vocabularies on different VERBMOBIL test sets.

Figure 3.11 shows the results. Obviously, TAZ vocabularies are not suitable for the VERBMOBIL domain: Batch, domain-specific VERBMOBIL vocabularies show consistently lower OOV rates than cross-domain TAZ vocabularies. Note too that the difference becomes larger for larger vocabularies. Only very frequent TAZ words provide a coverage comparable to incrementally built, domain-specific VERBMOBIL vocabularies. When less frequent words are included in the TAZ vocabularies, the cross-domain OOV rate decreases much more gradually than the domain-specific OOV rate. Interestingly, when all 700,000 words that occur at least two times in the TAZ corpus are included in the vocabulary, the OOV rate measured on different VERBMOBIL test sets is approximately 2.5 %. To achieve a similar coverage with domain-specific VERBMOBIL vocabularies, it is sufficient to include just 2,300 words.

3.6.2 Mixing Vocabularies

In the previous experiment the vocabularies of different corpora were kept separated. To test the usability of vocabularies that belong to a foreign domain, we were also interested in mixing vocabularies. Suppose that the entire VERBMOBIL corpus is not available, but only a portion of it, say 10 %. Can the coverage of this reduced VERBMOBIL vocabulary be improved by complementing it with words from the TAZ corpus? And further, how does the coverage of mixed vocabularies compare with larger domain-specific (VERBMOBIL) vocabularies?

Figure 3.12 shows the results of our investigation. For better visualization, the results of this experiment are plotted on both linear and logarithmic scales. The coverage of incremental domain-specific VERBMOBIL vocabularies is represented by the curve labeled VM/VM. The uppermost point in this curve is the average coverage of VERBMOBIL vocabularies created with 10 % of the VERBMOBIL corpus. When adding the most frequent words of the TAZ corpus to the vocabulary, the coverage of unseen VERBMOBIL utterances can be significantly improved, as can be seen in the curve labeled (0.1 VM + TAZ)/VM. As in the previous experiment, the biggest gain in coverage is obtained when adding high frequency TAZ words to the VERBMOBIL vocabulary. Notably, mixed vocabularies that include the most frequent TAZ words perform almost as well as domain-specific VERBMOBIL vocabularies of the same size (see the upper left part of the linearly scaled plot of Figure 3.12).

As the VERBMOBIL vocabularies become larger, supplementing them with TAZ words becomes less effective in reducing the OOV rate. This can be clearly seen in the curves (0.5 VM + TAZ)/VM and (0.9 VM + TAZ)/VM of Figures 3.12. They depict the coverage of TAZ vocabularies mixed with 50 % and 90 % of the VERBMOBIL corpus respectively. This outcome can be explained as follows: Words not included in large vocabularies tend to be low-frequency words. And low frequency words are more task dependent than high-frequency words, as discussed in [Hetherington, 1995a].

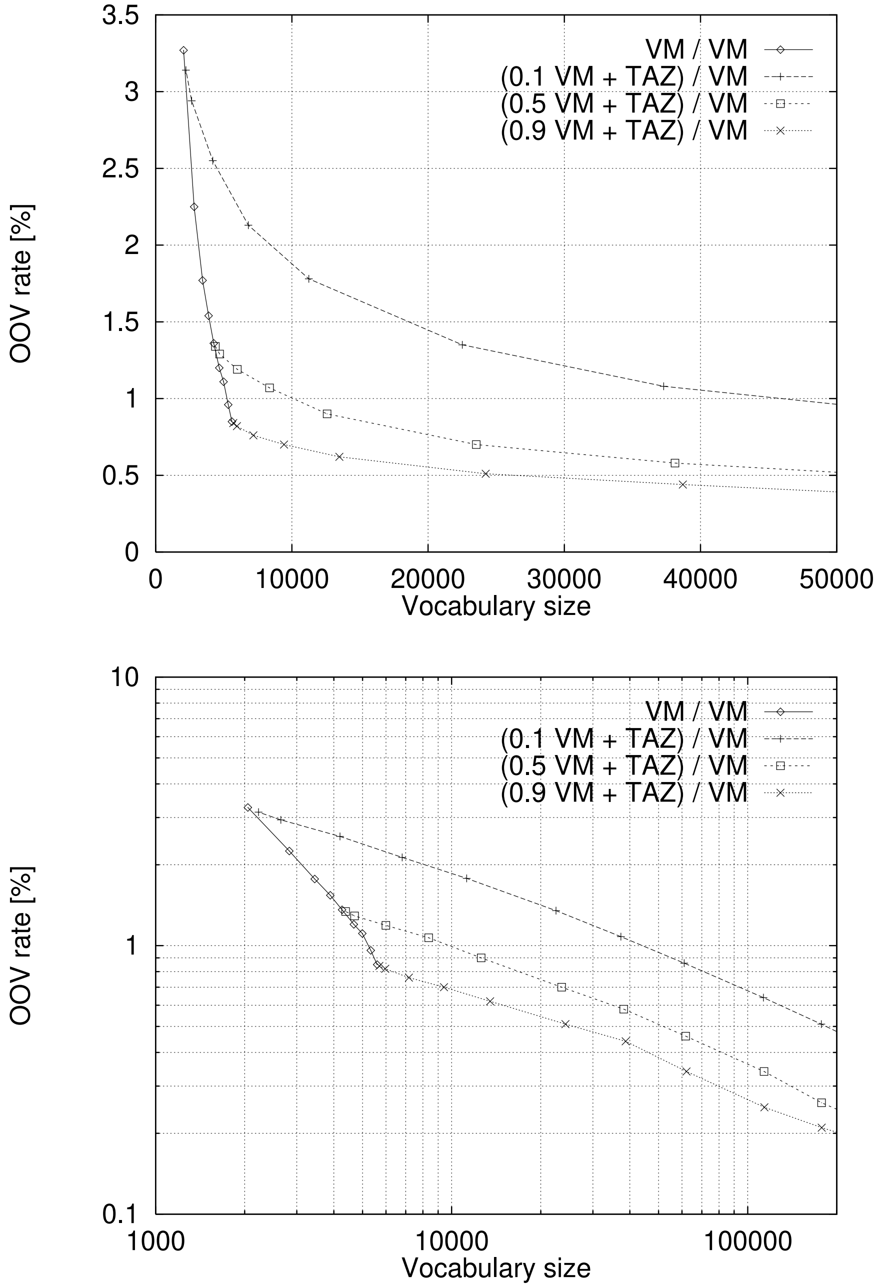


Figure 3.12: Coverage of domain-specific VERBMOBIL vocabularies versus mixed vocabularies from VERBMOBIL and TAZ. For better visualization, the results are presented both linearly and logarithmically. The steep heavy line on the left (VM/VM) represents the coverage of domain-specific VERBMOBIL vocabularies built incrementally. The three curves on the right show the coverage of TAZ vocabularies mixed with 10 %, 50 % and 90 % of the VERBMOBIL corpus respectively.

3.7 Summary

An analysis of the OOV-word problem was presented in this chapter. We first studied the relationship between corpus size and vocabulary size, and showed that the two parameters are highly correlated and strongly depend on the characteristics of the corpus being studied.

The coverage of different vocabulary sizes was also studied. It was shown that the OOV rate of unseen test material can be significantly reduced by increasing the size of the vocabularies. Yet even with very large vocabularies, the OOV rate remains very high, especially when measured at the turn (or sentence) level.

A classification of OOV words showed that proper names make up only a minor portion of all OOV words. Studying the acoustic properties of OOV words we discovered that they tend to be longer than in-vocabulary words, and that they are mainly made up of known acoustic sequences.

Finally, we investigated the coverage of "cross-domain" vocabularies, which have been created with a different corpus than they are tested. As expected, the OOV rate obtained under these conditions is much higher than the "intra-domain" OOV rate. Very frequent words, though, tend to be more domain-independent, and therefore exhibit a good coverage even in the difficult cross-domain scenario.

4 MORPHOLOGY-BASED METHODS

THE importance of German word compounding as a source of OOV words is evaluated in this chapter. We discover that compounds make up a large portion of all OOV words, and that many OOV compounds are composed of constituents that are already in the lexicon. In order to tackle this problem, we decided to integrate morphological knowledge into our speech recognition system. We do so by redefining the concept of "word" based on morphological considerations. In particular, we decompose compound words into their constituent parts, and use these as the lexical recognition units, instead of the compounds themselves. Later, we rejoin the parts to compound words in order to recover OOV compounds.

4.1 Introduction

In Chapter 3 out-of-vocabulary words were classified into three classes, namely "word fragments," "proper names," and "other OOV words." About two thirds of all OOV words belonged to the class "other." Inspection of this class showed that it contained a large number of inflected and compound words[1]. This is due to the fact that German morphology allows on the one hand to modify words by adding prefixes and suffixes, and on the other hand to concatenate words to form compound words. We realized that both morphological mechanisms are an important source of OOV words.

Further, it became apparent to us that the problem of OOV *words* is strongly related with the way *words* are defined. Popular opinion to date considers the written form of a language to be its primary form. From a linguistic point of view, however, rather the opposite is true. Speech recognition is concerned with

[1]The class "proper names" also contained many inflected, and especially compound words. Therefore, the distinction between "proper names" and "other OOV words" will not be made further. Only "word fragments" will be excluded from this study.

the classification of the *spoken* language, and is in no way dependent on the orthography of the language, or even the language having a written form. From the popular standpoint, words are normally defined in orthographic terms. This definition has the great advantage of being standardized, and is therefore frequently used in speech recognition for choosing lexical units. Furthermore, in languages where the set of morphological rules that govern the creation of word forms is rather limited, this approach might also be appropriate for speech recognition. But in languages with a very prolific morphology, such as German, the definition of a proper lexical unit might not be so straightforward and requires investigation.

We feel that morphological knowledge can be usefully employed in finding adequate lexical units, just as phonological knowledge is used to select acoustic models. In the next section we will briefly present some principles of German morphology that we will use in the course of this chapter. After that, we will review the research that has been previously done using morphology in speech recognition.

In our investigations we will first study the importance of derived word forms (created by morphological means). A frequency analysis of the Verbmobil corpus shows that inflections, derivations, and compound words make up a major fraction of German lexica. This conclusion is also valid for OOV words. Besides, we will show that many OOV compounds are composed of constituents that are already in the lexicon. This motivated us to experiment with recognition lexica based on the constituents of compound words rather than the compounds themselves.

In Section 4.5 all compound words will be decomposed into their constituents. Using these constituents as lexical entries, we will build a recognizer. Then, the recognition performance of this system will be compared with a traditional, word-based system. Since we first obtain a slight performance degradation by splitting compounds, we test different methods intended to automatically find a lexicon which is optimized for a given training material. This way a competitive performance is obtained with a lexicon which is better suited to the morphological characteristics of the German language.

Eventually, this morphologically-oriented lexicon will be employed for “recovering” OOV compounds. As mentioned above, many OOV compounds are made up of known constituents. We dubbed the process of recombining known constituents into one OOV compound “recovering.” This approach will be presented in Section 4.6.

4.2 Morphology and Speech Recognition

Morphology is the study and description of word formation [Bhatt, 1991], [Baumgärtner et al., 1973]. A **morph** is the smallest unit in a language that has a meaning, i.e. semantic content[2]. The morphology of a language defines a set of processes for creating word forms from a finite number of morphs. The mechanisms are often classified as:

1. Derivation
2. Inflection
3. Compounding

Derivatives are formed by adding prefixes, infixes, and suffixes to stem morphs. The derivation has a different meaning from the stem and often forms a new word class. A few examples are shown in Figure 4.1. There, hyphens are used to emphasize the boundaries between stem and derivation.

`plan-en`[3]	(to plan)	$\rightarrow$	`Plan-ung`	(the planning)
`gesund`	(healthy)	$\rightarrow$	`Gesund-heit`	(the health)
`dunkel`	(dark)	$\rightarrow$	`ver-dunkel-n`	(to darken)

Figure 4.1: Some examples of German derivatives. Hyphens are not part of the derivations, but are used to emphasize the boundaries. The English translation is given in parenthesis.

Inflections are also formed by adding prefixes, infixes, and suffixes to stems. As opposed to derivatives, though, the basic meaning of the inflected word remains unchanged. Inflections describe rather semantic relationships among words. Inflections are commonly divided into *conjugation* (for verbs) and *declension* (for nouns and adjectives). Tables 4.1 and 4.2 show examples of conjugation and declension respectively. Here, again, hyphens emphasize the boundary between stem and inflection.

[2] We consistently use the term "morph" throughout this chapter, because we work at the *realization* level. The *classification* implied by the term "morpheme" is irrelevant in this thesis. For more information on this subject see, for instance, [Bhatt, 1991], [Baumgärtner et al., 1973]

[3] The suffix "en" in `plan-en` is not a derivation but an inflection. The hyphen emphasizes the similarity of `plan-en` with `Plan-ung`.

	Present	Preterit	
ich	`leb-e`	`leb-te`	(I live/lived)
du	`leb-st`	`leb-test`	(you live/lived)
er, sie, es	`leb-t`	`leb-te`	(he,she,it lives/lived)
wir	`leb-en`	`leb-ten`	(we live/lived)
ihr	`leb-t`	`leb-tet`	(you live/lived)
sie	`leb-en`	`leb-ten`	(they live/lived)

Table 4.1: Conjugation is a frequent inflection form. In this example, the German verb `leb-en` (to live) is conjugated in present and preterit. In both tenses, four different word forms are originated. Here, hyphens do not belong to the inflected words, but are used to emphasize the boundaries beween stem and inflection. The English translation is given in parenthesis.

Compound words are formed by juxtapositioning word forms that result in a new word. There are basically two types of compounds: semantic and orthographic. In semantic compounds, which tend to be written together in many languages, the semantic content of the compound cannot be derived from the semantic content of its constituent parts, e.g. `jelly-fish` and `bath-room`. In orthographic compounds, on the other hand, the constituent parts retain their original semantic content, e.g. "coffee break," which in German is written "Kaffeepause" because German orthography requires both types of compounds to be written together.

In some cases, a linking morph (in German, usually "s" or "es") is inserted between the constituents, as in `Arbeit_s-essen` (business lunch) or `Jahr_es-tag` (anniversary). In addition, compounds can in turn become a constituent of a new compound word, as in `Sprach-erkennung` (speech recognition) and `Sprach-erkennung_s-technologie` (speech recognition technology). This is the origin of many extremely long words in German. A usual example of this phenomenon is the word `Donau-dampf-schiff-fahrt_s-kapitän` (captain of the Danube steamship company). Here, underscores are used to emphasize the presence of a linking morph.

German syntax allows many possible combinations of word classes in the construction of compounds. Table 4.3 shows a few examples [Bhatt, 1991], [Baumgärtner et al., 1973]. The most frequent combination is noun-noun, but all combinations of nouns and adjectives also tend to be frequent. In any case, the word class of the compound is determined by the class of its last constituent.

In this work, we will focus on compound words. One reason for this restriction

Case	Example	
Nominative	Da ist `der klug-e Hund`	(There is the smart dog)
Genitive	Wegen `des klug-en Hund-es`	(Because of the smart dog)
Dative	Ich gab es `dem klug-en Hund`	(I gave it to the smart dog)
Accusative	Ich sah `den klug-en Hund`	(I saw the smart dog)

Table 4.2: Declension is a frequent inflection form in German. In this example, the phrase `der klug-e Hund` (the smart dog) is declined in nominative, genitive, dative, and accusative. It can be seen that declension affects not only the noun, but also pronouns and adjectives. As in previous examples, hyphens are used to emphasize the morph boundaries, and the English translation is given in parenthesis.

is that word compounding is an unlimited source of OOV words. Inflections and derivations are formed with a limited number of prefixes and suffixes. This means that, although the full-form vocabulary size and consequently the number of potential OOV words may be very large, it is nonetheless finite. In compounds, on the other hand, the set of constituents in every position is unlimited, so that the number of potential OOV compounds is infinite. The other reason for focusing on compounds (as opposed to inflected words) is the intrinsic limitation of speech recognizers to recognize small sections of speech. This problem was first discussed by [Geutner, 1995], and will be explained in the next section.

4.3 Prior Work

Morphology has been largely ignored by speech recognition researchers. Only when large-vocabulary applications began to appear feasible did this subject get the attention it deserves. Furthermore, since morphology affects some languages particularly, like German, Dutch, Swedish, Finnish, Russian, Czech, Greek, etc. (but notably not English), it is not surprising that morphology was first applied outside the English speech recognition community. Pioneering work was done in Germany within the VERBMOBIL project.

Geutner presented a comparison between the German VERBMOBIL corpus (called the German Spontaneous Scheduling Task) and its English counterpart (the English Spontaneous Scheduling Task) [Geutner, 1995]. She shows that German is characterized by a much stronger vocabulary growth than English. In addition, German requires considerably larger full-word vocabularies than English to achieve a similar coverage.

Classes			Example	
Adjective	+	Adjective	`dunkel-rot`	(dark red)
Adjective	+	Noun	`Mittel-stand`	(middle class)
Adjective	+	Verb	`schwarz-sehen`	(to be pessimistic)
Noun	+	Adjective	`kreide-bleich`	(chalk-pale)
Noun	+	Noun	`Kranken-wagen`	(ambulance)
Noun	+	Verb	`ski-fahren`	(to ski)
Numeral	+	Noun	`Vier-beiner`	(four-legged animal)
Numeral	+	Numeral	`hundert-tausend`	(hundred thousand)
Preposition	+	Adjective	`über-groß`	(oversized)
Preposition	+	Preposition	`vor-bei`	(over)
Preposition	+	Verb	`über-sehen`	(overlook)
Preposition	+	Noun	`Nach-hilfe`	(tutorial help)
Verb	+	Adjective	`fahr-bereit`	(ready to drive)
Verb	+	Noun	`Geh-weg`	(sidewalk)
Verb	+	Verb	`schlafen-gehen`	(to go to sleep)

Table 4.3: Examples of word classes that can be combined to form compounds in German (taken from [Bhatt, 1991, Baumgärtner et al., 1973]). Hyphens and underscores are not part of the compounds. They only emphasize the boundaries. The English translation is shown in parenthesis.

In order to cope with the high German vocabulary growth rate, she proposes decomposing German word forms into their morpheme components. This increases the coverage of the German corpus to the "English level." Furthermore, the perplexity of the German language model is also drastically reduced[4]. Unfortunately though, the recognition performance declines considerably. This is due to the fact that many morphemes are very short and highly-confusable. To overcome this, Geutner uses a "not strictly linguistically-oriented decomposition." No details are given on the specific manner the vocabulary units are chosen, but in any case, it is a manual selection procedure, not an automatic. In order to compare the morpheme-based system with the baseline word system, the recognition results of the former are mapped into words. This mapping was performed by list look-up on the best hypotheses of the recognizer. On the word level, the morpheme-based system slightly outperformed the baseline system (0.7 % absolute increase in word

[4]Since perplexity figures depend on the vocabulary size, it is misleading to compare perplexities between language models with different vocabulary sizes. Perplexity tend to be lower with smaller vocabularies, biasing the comparison results towards the model with smaller vocabulary.

accuracy), with a roughly 30 % smaller vocabulary.

The results obtained are not spectacular, but interesting enough to inspire us to investigate the subject further. In particular, we are interested in the implications of mophological aspects on the problem of OOV words, and not just in obtaining a better recognition performance. In the work by Geutner, only the positive effect of smaller vocabulary units on the OOV rate was mentioned. Aside from this goal, our work represents an extension of Geutner's in at least three aspects: First, we search for *automatic* ways of adapting the vocabulary to the limitations of the recognizer (see Section 4.5.3). Second, the mapping from the sub-word level (morphemes) to the word level should occur on word graphs. And third, other mapping methods should be investigated, in order to overcome the limitations of list look-up (see Section 4.6). As regards to the improvement in recognition performance, we can say in advance that we obtained very similar results to Geutner's.

In [Spiess, 1995] an interesting method for handling compound words in the language model component of a speech recognizer was introduced. In German (as well as in Indoeuropean languages in general) the grammatically determining constituent of a compound word is the last one (the *tail* constituent). The other contituents (the *head* constituents) are modifiers of the tail. Based on this fact, Spiess proposes splitting compound words into their constituents, and then estimating the probability of word sequences depending on the position of compound constituents. The tails are predicted by their contexts, ignoring the compound heads; and the heads are predicted by their corresponding tails (which is equivalent to inverting the chronological order). Even though this approach has solid linguistic foundations, its implementation in the framework of an incremental speech recognizer is not trivial. Since in this paper no comparative experimental results are presented, we cannot judge if the necessary implementation effort is justified.

The SQALE project was the first major effort to evaluate speech recognition performance in a multilingual framework. Three state-of-the-art systems for American English ([Pye et al., 1995], [Lamel et al., 1995], and [Dugast et al., 1995]) were ported to British English, French, and German. The training and test materials had roughly the same size for all languages, and consisted of read newspapers (with the exception of the training set for the German acoustic models, which were taken from a different domain).

Originally, a vocabulary of 20,000 words was planned. For German, however, this vocabulary resulted in an OOV rate disproportionally high compared to English and French. (In [Dugast et al., 1995] an OOV rate of nearly 12 % is reported for German, compared to approximately 2 % for English and French.) In all three papers, German word compounding is viewed as the main reason for this poor lexical coverage. As a workaround, the German vocabulary size was increased to 64,000 words, which provided a coverage comparable to the other languages. This clearly shows the effect of word compounding on German vocabularies.

We presented our work dealing with compound word decomposition in [Berton et al., 1996a] and [Berton et al., 1996b]. These papers are the basis of the research discussed in the current chapter, and will be reviewed in the corresponding sections (Sections 4.4 to 4.6).

Compound word decomposition was also implemented within a Swedish-to-English spoken language translator [Carter et al., 1996]. As in our own work, compound words were splitted and the compound parts were incorporated into the lexicon, as well as in the language model. For evaluation purposes, the compound parts were reunited after recognition. As opposed to our work, where constituents were rejoined on word graphs, rejoining was performed on a single word string, which is simpler, but also more error-prone. The results obtained are comparable to our own: Slight improvements of recognition performance (below 0.5 %) could be achieved. Furthermore, compound word decomposition also enhanced the end-to-end performance of the system.

The work presented in [Lüngen et al., 1996] concentrates on so-called split compounds. These are compound words which were uttered in two separate sections, often divided by (filled) pauses. The authors found that splitting takes place mainly at morphological boundaries, like in `April-<Pause>-hälfte` (half of April). Based on this observation, they propose three different templates to model split compounds, depending on the characteristics of the segment inserted between compound parts.

To investigate this phenomenon, word graphs of Verbmobil utterances containing split compounds were generated using the speech recognizer described in [Hübener et al., 1996]. Then the word graphs were searched for split-compounds patterns [Lüngen et al., 1996]. Whenever such a template was found (provided, of course, that the compound parts are in some way contained in the recognition lexicon), they were replaced by the rejoined compound. Since the number of split compounds present in the test set is rather small, word accuracy differences between the word-based and contituent-based systems are not statistically significant. Besides, word accuracy was unfortunately only measured on word graphs. The word accuracy of the best sentences would have been very instructive, particularly considering the tendency of the templates to overgenerate. Nevertheless, this approach is interesting, and can easily be extended to deal not only with split compounds, but with all compounds.

This method is similar to our work [Berton et al., 1996a, Berton et al., 1996b] with respect word graph searches. But it is also a more general approach in two senses. First, the templates allow rejoining compound constituents which are not directly connected by a common node in the word graph. Our lexical search finds only connected arcs in the graph structure. Second, the arc labels used for the lexical search are not the inflected compound constituents, but their stem forms. Consequently, the algorithm can theoretically rejoin incorrectly inflected hypotheses to a correct compound word, as illustrated in Figure 4.2. My feeling is that this advantage in flexibility could greatly increase the number of compounds hypothesized when all possible compounds, and not only split compounds are considered. Clearly, more investigation is necessary in this subject.

• Spoken utterance:	Arbeit_s-<Pause>-amt
• Recognizer output:	arbeit-en Amt
• After rejoinig:	Arbeit_s-amt

Figure 4.2: Example of the method presented in [Lüngen et al., 1996] to rejoin split compounds. The German phrase Arbeit_s-amt (job center) was uttered with a silence between constituents. The recognizer did hypothesize the right compound stems, but wrong inflections. The algorithm reconstructs the correct compound.

Hwang tries to find lexical units that are more suited than words for speech recognition [Hwang, 1997]. He begins by splitting all words in the vocabulary into their morphemes (or syllables). After that, he rejoins the morpheme pairs (or syllable pairs) with the highest bigram frequency. This process is repeated until the highest bigram frequency is below a given threshold. This way, the vocabulary size can be adjusted to a chosen operational point between the number of distinct morphemes (or syllables) and the number of words. Varying the vocabulary, also varies the perplexity of the corpus. Since perplexity depends on the vocabulary size, some kind of normalization is necessary in order to compare perplexities among different sized vocabularies. In this paper, perplexity measurements were normalized over the average number of phonemes per symbol (words, morphemes, or syllables).

The experiments were conducted on the Korean Spontaneous Speech Database, the Korean counterpart of VERBMOBIL. The results presented are somehow contradictory. Training set perplexities drop from a high level for small vocabularies to a low level for large vocabularies. As the vocabulary grows larger, the morpheme perplexity converges to the value of word perplexity. Test set perplexities behave more erratically, beginning with low values for small vocabulary sizes, and growing far above the level of word perplexity. Hwang argues that this inconsistence is caused by overoptimizing the vocabulary to the training set. Recognition experiments were also disappointing: The traditional, word-based system outperformed both morpheme and syllable recognizers.

From the point of view of lexical optimization based on morphological units, this work is similar to ours [Berton et al., 1996b]. An important difference, though, is the fact that Hwang does not consider word boundaries when optimizing the vocabulary[5]. Therefore, the lexical units gained in this process can no longer be unequivocally mapped into the corresponding words. If words are the defined interface between the speech recognizer and other system modules, this method cannot be applied.

4.4 Analysis

In order to study the importance of the morphological mechanisms explained in Section 4.2, namely derivation, inflection, and compounding, a frequency analysis on part of the VERBMOBIL corpus was performed. For this study, we chose the subset of the corpus from which information on morphological word segmentation was available at the time our work began. This subset includes the training and test sets defined for the VERBMOBIL Evaluation in 1995 [Lehning, 1995][6]. (See Section 3.3.2 for a description of the VERBMOBIL corpus.)

The word segmentation was extracted from "The Bielefelder Lexicon Database" [Bleiching, 1995], which was provided by our project partners from the University of Bielefeld, Germany. There, derivatives, inflections, and compound constituents are marked. Since this database is first automatically segmented and then hand-corrected, a high degree of correctness is guaranteed.

Figures 4.3 and 4.4 summarize the results of the frequency analyses of the training and test materials, respectively. On the left, it can be seen that roughly one fourth of all words in the corpus (running text)[7] are inflected, and about one in eight words is a compound. These figures are consistent for the training and test sets. On the right side of the pictures, the proportion of inflections, derivations and compounds in the vocabulary is depicted. It is obvious that words affected by these morphological mechanisms make up the vast majority of the vocabulary: Over half of the words in the vocabulary are inflected, and approximately one third of all words are compounds. We conclude that even though the number of inflected words and compound words in the running text is relatively low, they

[5]Word boundaries are not as important in Korean as they are in Indoeuropean languages.

[6]The vocabulary used in this study is slightly larger than the "official" vocabulary defined for the Evaluation in 1995. The 29 symbols of the German spelling alphabet, as well as 14 additional words were appended to the official vocabulary of 3,307 words.

[7]Frequency figures based on corpus counts are often referred to as *token frequencies*. Frequency counts based on the vocabulary are termed *type frequencies*.

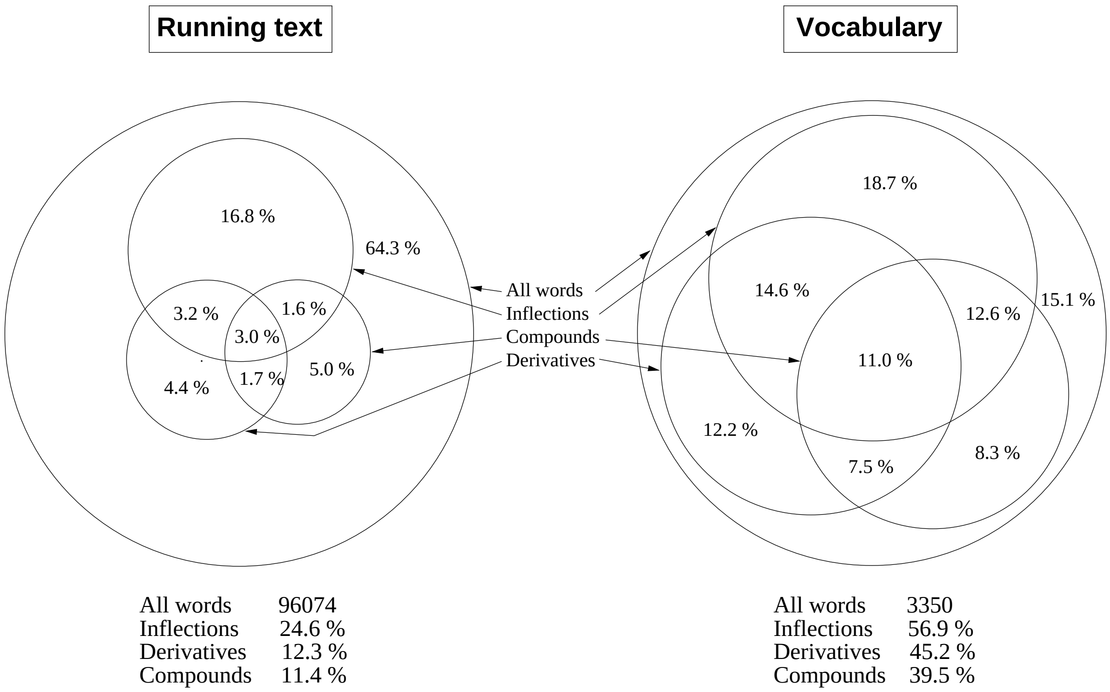


Figure 4.3: Proportion of inflections, derivations and compounds in the training material of the VERBMOBIL corpus used in this chapter. On the left side, the percentages are based on the corpus (Running text). On the right, percentages of the vocabulary are shown.

make up a major fraction of the vocabulary. From this observation, we infer that very frequent words tend to be unaffected by morphological mechanisms, whereas inflected words and compounds tend to be less frequent.

OOV Words

Since, as mentioned before, our main goal in this chapter is to recover OOV words by morphological means, we also analyzed the OOV words of the test set. For this experiment, we consider only "correct" OOV words, that is, word fragments are excluded from the statistics[8]. Figure 4.5 shows the results. In this case, there are only minor differences between the figures obtained in the weighted and unweighted analyses (on the left and right side, respectively), because most OOV words occur only once. It is interesting to note that over 40 % of all OOV words

[8]Word fragments make up roughly 1.1 % of the test set. This translates into a share of 30 % of all OOV words.

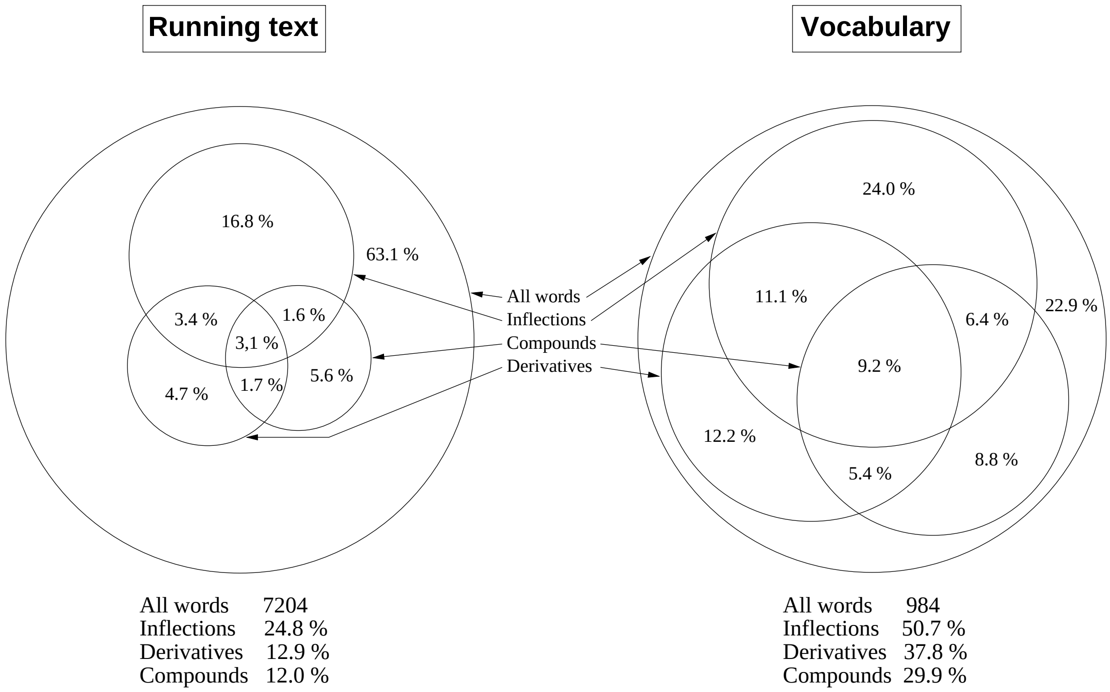


Figure 4.4: Proportion of inflections, derivations and compounds in the test material of the VERBMOBIL corpus used in this chapter. On the left side, the percentages are based on the corpus (Running text). On the right, percentages of the vocabulary are shown.

are compound words.

We further analyzed the constituents of all OOV compounds. To our surprise, we found that over 50 % of all OOV compounds are formed with components which are already in the vocabulary[9]. In other words, roughly every fourth OOV word (more precisely 23.4 %) consists entirely of known constituents. In addition, over 90 % of all OOV compounds contain at least one constituent which is already in the vocabulary. Figure 4.6 illustrates this graphically.

Compound Words

In this work we are especially interested in compound words. We therefore studied two aspects of word compounding, which will be used in Section 4.6 when

[9] We refer to word forms and *not* to word stems. The constituents of the OOV word `Weih-nacht_s-ferien` (Christmas holiday) for example, which are `weih, nachts,` and `ferien`, were already in the lexicon.

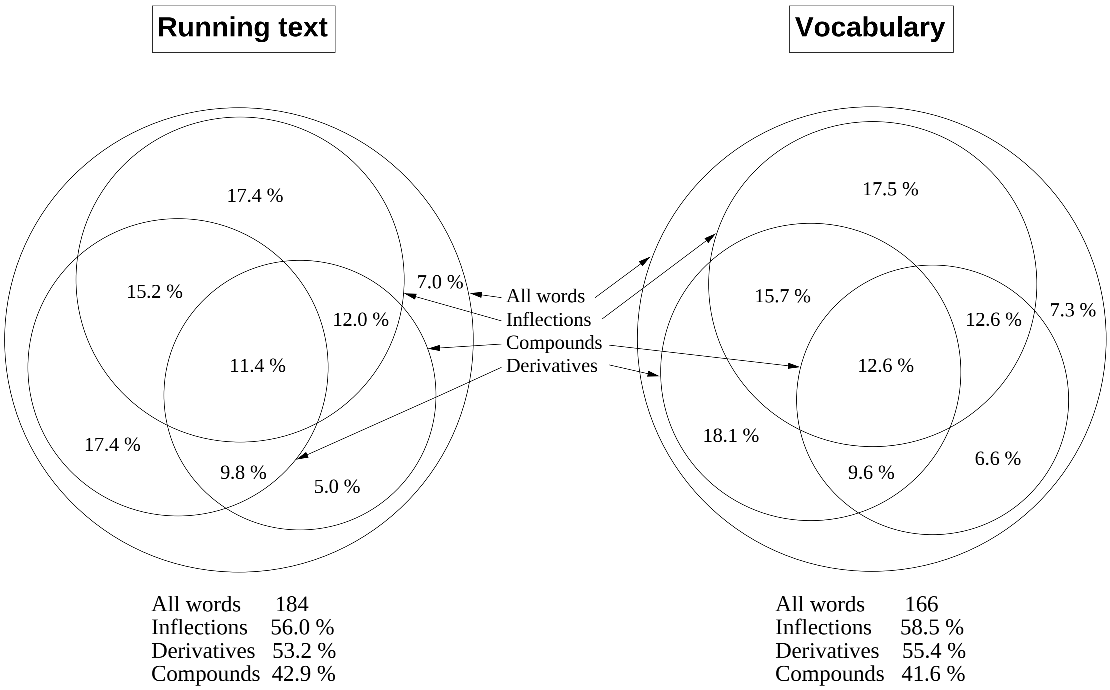


Figure 4.5: Proportion of inflections, derivations and compounds in the OOV words in the test material of the VERBMOBIL corpus used in this chapter. On the left side, the percentages are based on the corpus (Running text). On the right, percentages of the vocabulary are shown.

describing how to recover OOV compounds. First, we analyzed the number of constituents that form compounds word in the training corpus. Table 4.4 shows that the great majority of compounds have either two or three constituents.

A further aspect we are interested in is the constitution of compound words in terms of the syntactic classes of their constituent parts. The classes we used for this study were also taken from "The Bielefelder Lexicon Database" [Bleiching, 1995]. There, twelve syntactic classes are differentiated:

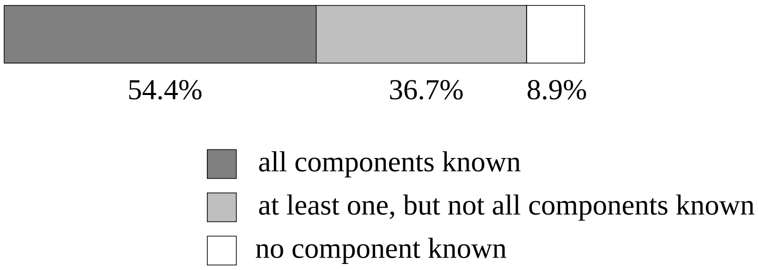


Figure 4.6: Proportion of OOV compounds with known constituents.

Number of constituents	λ_N
$N = 2$	81.0 %
$N = 3$	18.5 %
$N = 4$	0.4 %
$N > 4$	0.1 %

Table 4.4: Distribution of the number of constituents of compound words in the training corpus. λ_N is the percentage of all compounds having N constituents.

NA: not-inflected adjective

IA: inflected adjective

A: adverb

D: determinant

I: interjunction

C: conjunction

N: noun

PD: preposition with determinant

NU: numeral

P: preposition

PN: pronoun

V: verb

(The capital letters identify the classes in Figure 4.7.) We also manually tagged the classes of all constituents that are not by themselves words, because they were not included in the database. At the same time we introduced an additional class:

Ns: noun with linking morph

which we found beneficial.

All compound words in the training corpus were decomposed into their constituents, and the conditional probability of class C_l being followed by class C_r

was computed for all constituents, as follows:

$$\mathrm{P}(\mathcal{C}_r \mid \mathcal{C}_l) = \frac{\mathrm{Count}(\mathcal{C}_l, \mathcal{C}_r, w \text{ is a compound})}{\mathrm{Count}(\mathcal{C}_l, w \text{ is a compound})} \tag{4.1}$$

Figure 4.7 depicts the conditional probabilities of all class pairs. It is evident that some classes have a special affinity to other classes. For example, if a "left" constituent belongs to the class `Ns` (Noun with linking morph), it will almost always combine with another noun (class `N`). Other classes have a more evenly distributed conditional probability of succeeding classes, such as adverbs or prepositions (classes `A` and `P`).

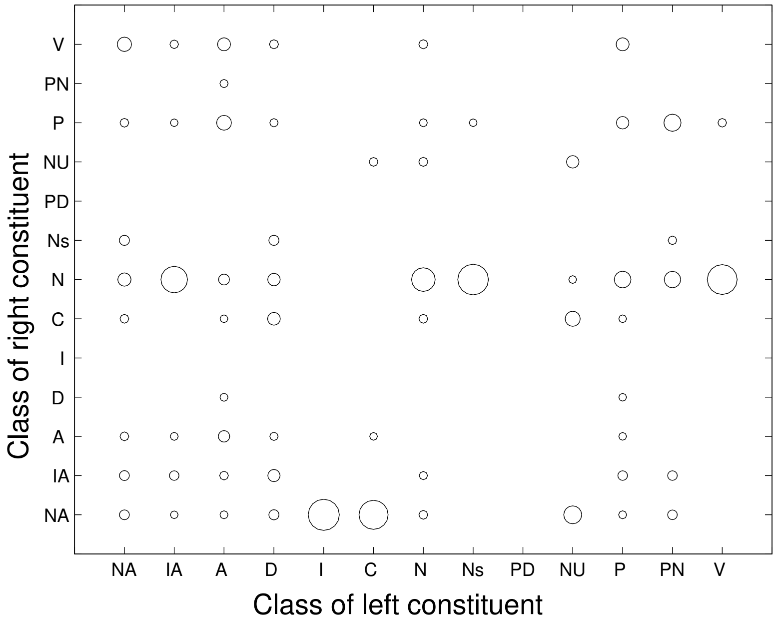


Figure 4.7: Conditional probabilities of constituent class pairs. The classes of "left" constituents are shown on the abscissa, and the classes of "right" constituents on the ordinate. The radius of the circle at the intersection of class $\mathcal{C}_l$ and class $\mathcal{C}_r$ is proportional to the probability $\mathrm{P}(\mathcal{C}_r \mid \mathcal{C}_l)$.

At the beginning of this section we saw that compound words make up about one third of the vocabulary in the VERBMOBIL corpus. In order to study the importance of compounds in very large vocabularies, we also studied the database from the TAZ newspaper, which was introduced in Section 3.3.3. We constructed several vocabularies, each one consisting of the n most frequent words in the TAZ corpus, and then determined the number of compounds in each vocabulary.

For this, the program *Mikrogrammatik* by Stall [Mangold et al., 1978] was used, which decomposes any German word along phonological lines and generates as a byproduct a morphological decomposition.

Figure 4.8 shows that as the vocabulary grows, the proportion of compounds increases from approximately one third to approximately two thirds of the vocabulary.

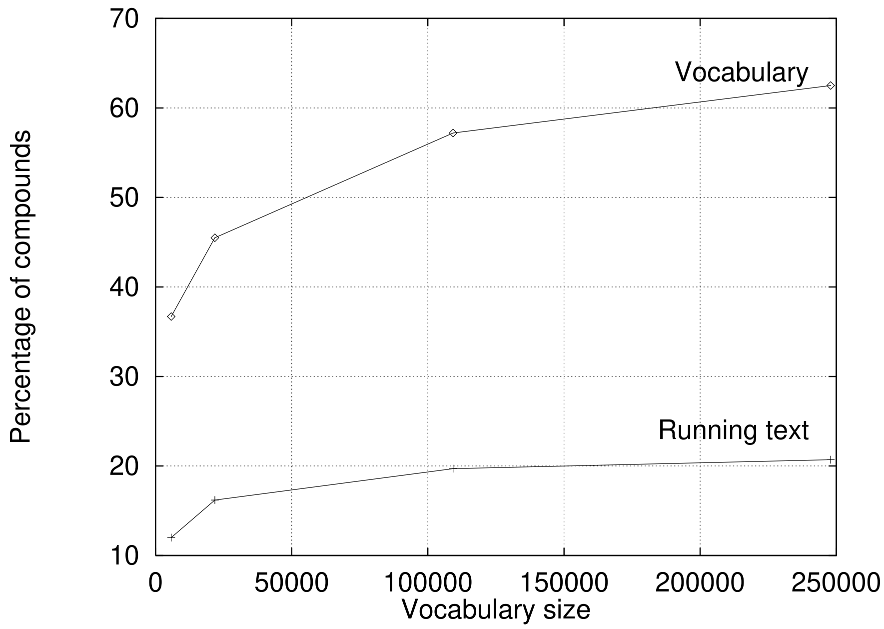


Figure 4.8: Percentage of compound words in the TAZ database. The upper curve shows the proportion of compounds in the the vocabulary and the lower curve the proportion in the corpus.

4.5 Morphologically-Oriented Lexical Recognition Units

How can morphological knowledge be integrated into our speech recognition system? Since the experiences we have collected so far in this chapter are related to the structure of words, the obvious way of integrating morphological knowledge is at the lexical level. In this approach, the recognizer lexicon contains not only words, but also morphologically-oriented lexical units, such as morphs, or compound constituents.

A lexicon based on morphs would significantly reduce the vocabulary size, and at the same time the OOV rate. Unfortunately, though, we suspect that due to the shortness of many (in German most) inflectional and derivational prefixes and suffixes, it would also reduce the performance of the acoustic recognizer, as reported in [Geutner, 1995]. Since experiments (presented in [Berton, 1996]) confirmed our suspicions, we abandon this approach at this point. This experience taught us that reducing the vocabulary size cannot be a goal in itself when trying to optimize the lexical units used for recognition. It is also necessary to consider the limitations of the speech recognizer.

As opposed to morphs, compound word constituents are often sufficiently long for good acoustic discrimination. Therefore, we believe that the performance degradation obtained with morphs does not apply to compound words. On the other hand, the benefits of reducing vocabulary size and the OOV rate still hold, as was shown in Section 4.4. For these reasons we decided to focus only on compound words. In the next sections we present our approach based on compound constituents as lexical recognition unit.

All the experiments were performed on the test set defined for the VERBMOBIL Evaluation in 1995. Also the acoustic and language models were trained with the "official" training material, as defined in [Lehning, 1995]. These training and test sets were chosen because morphological information was available for them at the time this study began. The subset is roughly one third of the whole VERBMOBIL corpus, as introduced in Section 3.3.2, so the baseline system used in this section is suboptimal in performance. But since this limitation is also valid for the systems being compared to the baseline, it is not relevant to this chapter.

4.5.1 Constituent Recognition

All compound words in the VERBMOBIL lexicon were first decomposed into their constituent parts. This significantly reduced both the vocabulary size and the OOV rate (see Table 4.5). The OOV rate reduction is due to the fact that most constituents of compound words are already in the lexicon, as explained in Section 4.4.

System	Vocabulary	OOV rate[10]	PP[11,12]	WA[13]
Baseline	3350	3.65 %	116	62.7 %
Constituents	2561	2.58 %	77	64.0 %
Δ	-24 %	-29 % (rel)	-34 %	+1.3 % (abs)

Table 4.5: Comparison of the baseline (word-based) system with the constituent-based system. Vocabulary size, OOV rate, perplexity and word (or constituent) accuracy of the best hypothesis are shown.

Next, a bigram language model based on constituents was trained and tested. The perplexity of this language model turned out to be considerably lower than the word-based one. Since perplexity measurements in models with different vocabulary sizes tend to bias the results towards the model with the smaller vocabulary, we attempted to quantify the effect of the factors involved. The perplexity reduction is the result of a smaller vocabulary, a lower OOV rate, and the better symbol inventory. The effect of the smaller vocabulary was estimated by reducing the vocabulary size of the word-based model to the number of symbols contained in the constituent-based model (where a symbol is either a word or a constituent). The effect of OOV words was estimated by not considering bigrams that involve OOV words in the computation of the perplexity. These two factors together account for approximately 20 % of the relative perplexity reduction (or 8 perplexity points, in absolute terms). We conclude that the constituent-based language model is in effect significantly better than the word-based one.

Afterwards, the recognizer was run with both the word-based and the

[10] As opposed to the analysis presented in Section 4.4, word fragments are now included when computing the OOV rate. When word fragments are excluded from the computation, the OOV rates are 2.55 % and 1.62 % for the baseline and constituent-based system, respectively.

[11] We report perplexity measurements that include transitions involving OOV words. If these transitions are excluded, the bigram perplexity of the baseline system is 95, and the perplexity of the constituent-based system is 67.

[12] As mentioned elsewhere in this thesis, perplexity measurements depend strongly on the treatment of the noise marker, because it is a very frequent event in VERBMOBIL. Since we observed that explicitly modeling noises in the language model is beneficial to the recognizer, we preserve them in the training text and include it in the vocabulary of the language model. But for perplexity computations we prefer to ignore it, because its very high frequency of occurrence tend to distort the figures.

[13] As mentioned before, in this study we used only approximately one third of the VERBMOBIL corpus for training the acoustic and language models. This subset was chosen because morphological information was available at the time the study began. When using the whole VERBMOBIL corpus for training, our system performs significantly better: On the same test set we measured a word accuracy of 71.1 %, that is an improvement of 8.4 percent points.

constituent-based vocabularies. Not only the best hypothesis was computed, but also word graphs, which were used for post-processing and will be discussed in the next section. The results illustrated in Figure 4.9 show a slight superiority of the constituent-based system over a wide range of word densities. No difference in run time could be measured between word-based and constituent-based systems. This contradicts results reported in [Geutner, 1995], where the constituent-based system required significantly less computation. This difference is due to the different lexicon structure: Whereas the recognizer used in [Geutner, 1995] has a linear lexicon structure, our lexicon is organized as a tree which efficiently reduces the redundancy present in the vocabulary.

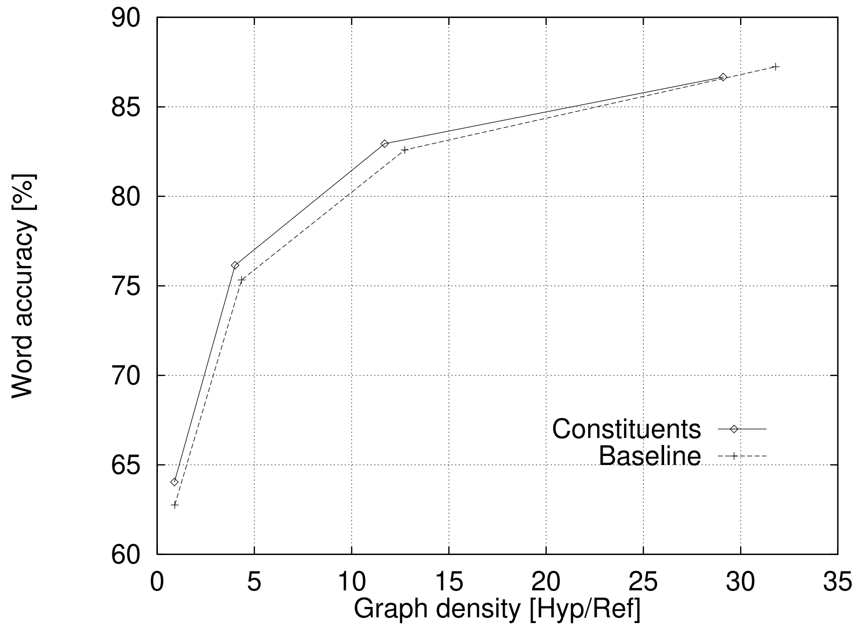


Figure 4.9: Performance comparison of the baseline (word) system with the constituent-based system over a wide range of graph densities. The points on the left show the word accuracy of the best hypotheses, and are the same as the figures shown in Table 4.5.

4.5.2 Rejoining Constituents

Since speech recognizers are usually embedded in an application and have to communicate with other system modules, the output of the recognizer has to be compatible with the input expected by these modules. In VERBMOBIL, the various parsers used to interpret the utterance are based on the classical definition of German orthography. The output of the constituent-based recognizer, though, contains (non-compound) words *and* compound word constituents. The latter have to be rejoined to words that can be processed by the parsers. In addition, the word level provides a common basis to compare the baseline system and the constituent-based system.

Rejoining compound constituents can be based on a lexicon, on rules, or on probability criteria. In the lexical method, compound constituents are joined if they appear in a list of permitted compound words. This method has the advantage of being especially simple to implement, but is bound to rejoin only known compounds. On the other extreme, rule-based systems are not limited to a list of permitted compound words, but require an extensive knowledge base in order to model word compounding sufficiently well. Probabilistic methods provide a compromise between feasibility and ease of implementation on the one hand, and generalization power on the other.

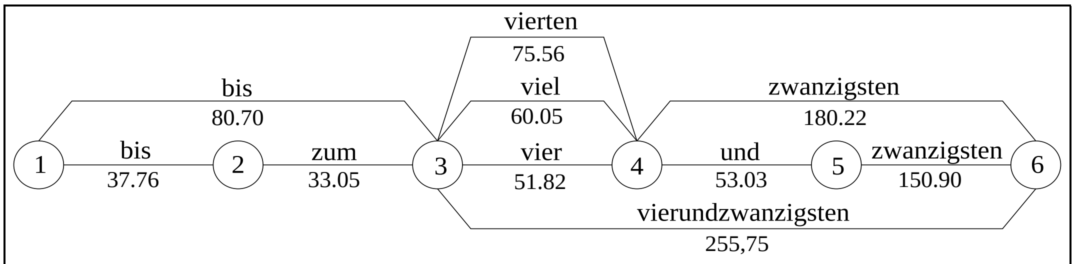
In the first experiments, our intention was to gain some experience and evaluate the strengths and weaknesses of rejoining constituents. So we opted for the simplest rejoining method, namely the lexical search. When we attempt to tackle the problem of OOV compounds in Section 4.6, we will use a probabilistic approach.

We begin with the hypotheses generated by the constituent recognizer presented in the last section. We chose word graphs with a relatively high word density (≈ 30 hypotheses per spoken word) because larger graphs are more likely to contain the right hypotheses. When a compound candidate is detected in the word graphs, an additional edge is inserted into the graph, yielding the compound word hypothesis. As the algorithm traverses the graph, every candidate can, in turn, become part of a longer compound, so that compounds of any length can be recovered. The acoustic scores of the constituents are simply added[14]. For a more detailed description of the procedure we refer the reader to [Berton, 1996]. Figure 4.10 illustrates an example of this method.

After inserting the compound candidates, the dynamic-programming pruning algorithm described in Chapter 2 was applied using a word-based language model to extract the best sentence. The whole-word recognition system (our baseline) yielded a 62.2 % word accuracy[15] after graph pruning. The constituent-based

[14]Since acoustic scores are logarithmic probability-like measures, adding scores correspond to multiplying probabilities.

[15]This is different from the word accuracy presented in Section 4.5.1, because this time the best sentence was extracted from an intermediate (already reduced) word graph. This procedure was chosen to allow a fair comparison with the system based on compound constituents, which also depends on intermediate word graphs.



- Spoken utterance: *bis zum vierundzwanzigsten*
- Recognizer output: *bis zum vier und zwanzigsten*
- After postprocessing: *bis zum vierundzwanzigsten*

Figure 4.10: Example of word graph processing. The German phrase `bis zum vier-und-zwanzigsten` (up to the twenty-fourth) was uttered. The constituent-based recognizer hypothesizes `bis zum vier und zwanzigsten`. On the word graph, the compound `vier-und-zwanzigsten` is rejoined to `vierundzwanzigsten`.

system obtained an accuracy of only 60.7 %. The 1.5 % deterioration in word accuracy was caused mainly by the worse recognition of the individual components of the compound words, as shown in Table 4.6.

System	WA (all words)	WA (non-comp.)	WA (compounds)
Baseline	62.2 %	61.0 %	82.5 %
Rejoined constituents	60.7 %	60.4 %	72.8 %

Table 4.6: Performance comparison of the baseline (word-based) system with the rejoined-constituent system. The overall word accuracy was also separately measured for compound and non-compound words.

This poorer performance is due to the fact that contextual information used extensively in the baseline system is ignored in the constituent-based system. On the one hand, the acoustic information provided by context-dependent models is ignored at constituent boundaries (because we do not use cross-word acoustic models). Whereas the word `Absprache` (agreement), pronounced `/apSpra:x@/`, is modeled by the model succession `[--a][-ap][apS][pSp][Spr][pra:][ra:x]-[a:x@][x@-][@--]` in the word-based system, the word components `Ab sprache` are modeled by the much less specific model succession `[--a][-ap][ap-][p--]-[-S][-Sp][Spr][pra:][ra:x][a:x@][x@-][@--]` in the constituent-based system. On the other hand, the stiff constraints hard-coded into the word-based lexicon are only partly compensated for by the relaxed constraints imposed on compound components by the language model.

To make things worse, errors in the constituent-based word graph contribute more to the overall error rate after the rejoining procedure than before it. This is the case whenever one of the constituents is not hypothesized in the word graph. In that particular situation, the compound word containing the missing constituent cannot be restored by the search algorithm.

4.5.3 Vocabulary Optimization

As opposed to the baseline system, where no compounds are decomposed, in the "rejoined-constituent" system just described, all compounds are decomposed. The results, however, are not satisfactory. We now examine three methods for automatically selecting a subset of compounds to be decomposed, in order to lessen the drawbacks discussed above.

Number of Phones

Some of the compound words in the text contain very short components, which, like inflectional and derivational morphs, tend to be easily misrecognized. This is particularly true for 2-phone constituents because of coarticulation effects, as in `Ab-sprache` (agreement). In our first optimization approach we therefore suppressed the decomposition of words containing constituents with less than a given number of phones (see Figure 4.11). This experiment shows that word accuracy increases almost linearly with vocabulary size but never exceeds the baseline word accuracy.

Compound Word Frequency

Some compounds occur so frequently in the training that their context can be trained well enough in the language model. The advantage of a constituent-based system, namely the higher frequency of each entry, decreases for such words. We therefore decided that very frequent words should not be decomposed. The results are shown in Figure 4.12. This optimization criterion improves word accuracy by 2 % with a vocabulary increase of 4 % compared to the "rejoined-constituent" system. This result is very promising because the vocabulary is 20 % smaller and word accuracy increases by 0.5 % over the baseline system.

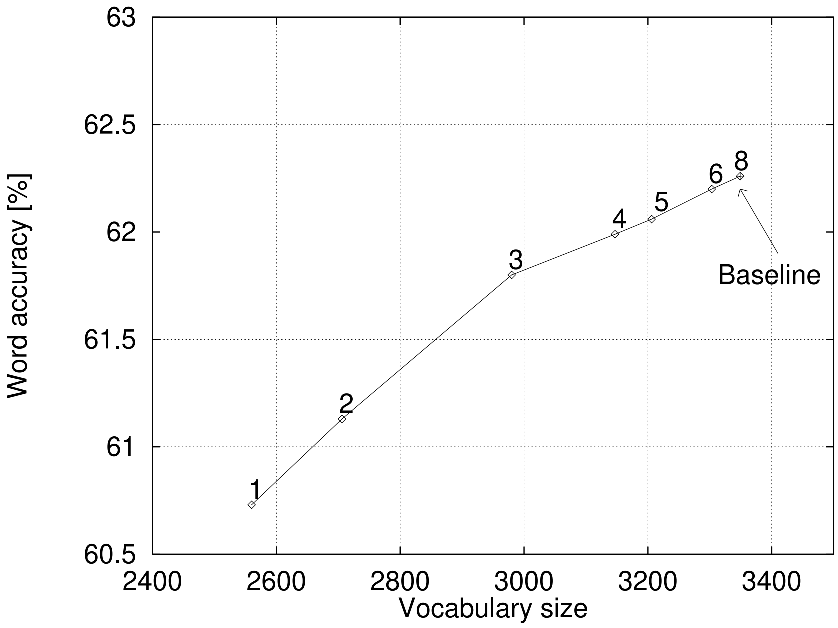


Figure 4.11: Vocabulary optimization using a phonological criterion. Compound words are not split whenever the number of phones in the constituents is less than or equal to a given threshold. The point marked with a 1 corresponds to the "rejoined-constituent" system, where all compounds are decomposed. The point marked with an 8 is the baseline (word-based) system.

Constituent Interdependence

It was found that some constituents occur only in conjunction with a very limited number of other constituents. They cannot be freely combined and thus do not increase performance if they are decomposed. We defined the **C**onstituent **i**nter**d**ependence *CID* is as the ratio of the bigram count $\mathrm{Count}(c_1, c_2)$ to the average of the number of times the constituents c_1 and c_2 have been seen in the training:

$$\mathrm{CID}(c_1, c_2) = \frac{\mathrm{Count}(c_1, c_2)}{\sqrt{\mathrm{Count}(c_1)\mathrm{Count}(c_2)}} \tag{4.2}$$

The constituent interdependence *CID* is larger for constituent pairs that tend to occur together. Optimization based on it consists of not decomposing those compounds for which their *CID* exceeds a given threshold. The system was tested for several *CID* thresholds, and the system performance could be slightly improved over the baseline system (see in Figure 4.13).

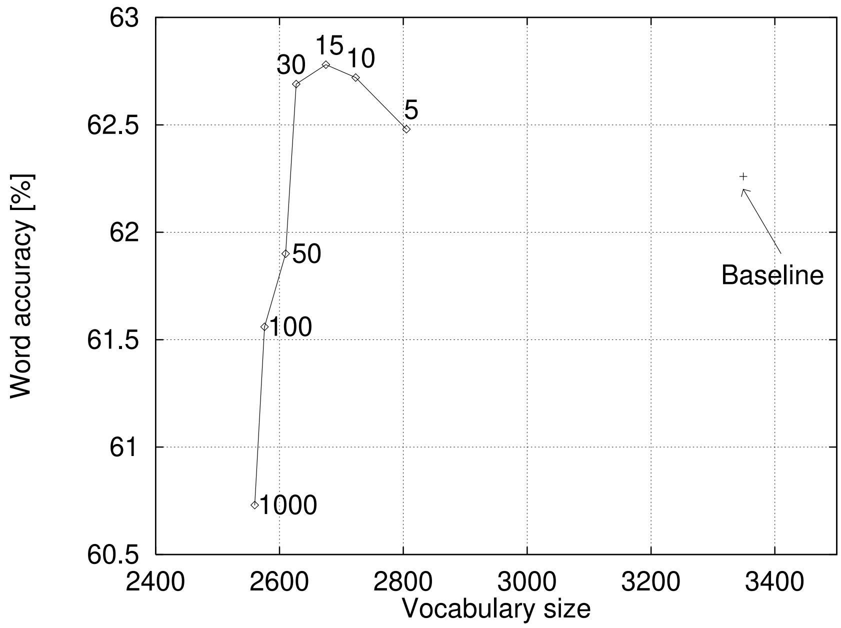


Figure 4.12: Vocabulary optimization using a frequency criterion. Compound words are not splitted if they occur more frequently than a given threshold in the training corpus. The best results were obtained with a threshold of `15` occurrences, though this number is clearly dependent on the magnitude of the training corpus.

A constituent c_1 was dubbed *bound*, if the bigram count equaled the absolute count of the constituent:

$$\mathrm{Count}(c_1, c_2) = \mathrm{Count}(c_1) \tag{4.3}$$

This means that the constituent occurs exclusively in a given context, e.g. in the word `Mon-tag` (Monday), the constituent `Mon` is bounded. We tested the system for the case that no compounds with bounded constituents were decomposed. Compared with the non-optimized constituent-based system, word accuracy improved by 0.5 %, and the lexicon was reduced by 2 % (see Figure 4.13).

Summary of Results

Figure 4.14 summarizes the results obtained by the vocabulary optimization techniques presented above. The most effective method was to suppress decomposition of very frequent compounds. Using this optimization approach we could slightly

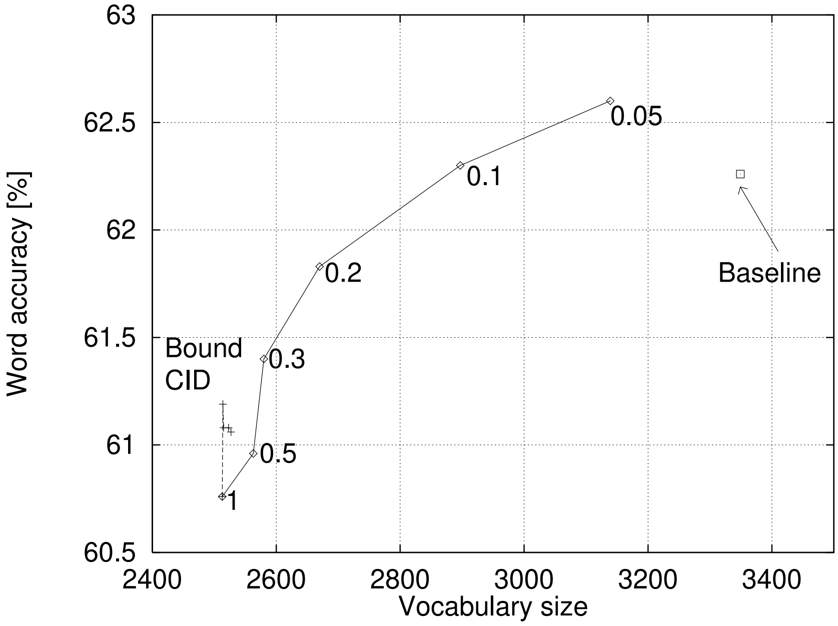


Figure 4.13: Vocabulary optimization using a constituent interdependence criterion. Compound constituents are not split if they tend to occur together in the training corpus. The points in the plot represent different *CID* thresholds.

outperform (by 0.5 % word accuracy) the baseline system with a vocabulary reduction of 20 %. This improvement in accuracy is comparable to that obtained by other research groups [Geutner, 1995], [Carter et al., 1996]. A combination of the three optimization techniques yielded no further gain in performance.

4.6 Recovering Out-Of-Vocabulary Compounds

In Section 4.4 we saw that the components of over half of all OOV compounds and almost one fourth of all OOV words are already in the lexicon. Due to the rejoining approach based exclusively on a list of permitted compound words, none of these OOV compounds was correctly hypothesized in the systems presented above. In order to find OOV compounds in the word graph and to be able to rejoin their constituents (a process we dubbed "recovering OOV compounds"), we applied two probabilistic methods.

In both methods, we performed a search on word graphs containing separated compound constituents. A pair of constituents is rejoined to a compound candi-

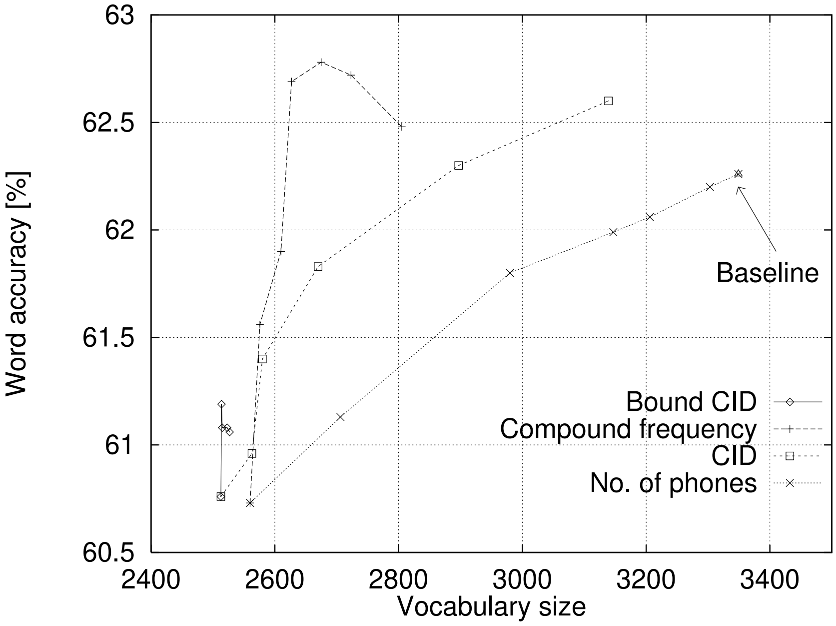


Figure 4.14: Comparison of the optimization strategies.

date either if it is contained in the list of in-vocabulary compounds or if it meets certain criteria aimed at recovering OOV compounds, as will be explained below. Just as before, every candidate can, in turn, become part of a longer compound, so that OOV compounds of any length can be recovered. At the end, the most likely sentence is extracted from the graphs, as explained in Chapter 2. All results reported here refer to these most likely sentences.

4.6.1 Head and Tail Probabilities

We observed that the constituents of OOV compounds which already occurred in the training corpus tend to do so as part of other compounds. Therefore, we want to investigate if the identity of a symbol (word or constituent) provides enough information to decide if this symbol belongs to a compound word or is a word by itself. In other words, our goal is to estimate the probability that a given sequence of constituents is in fact a compound word.

In order to estimate this probability, we make use of the statistical properties of compound word *heads* (first constituent) and *tails* (remaining constituents)[16].

[16] Here we use a different definition of compound heads and tails as the one introduced in [Spiess, 1995]. Whereas the definition by Spiess has linguistic foundations, ours is more convenient for sequentially processing constituent sequences.

The head and tail probabilities were defined as the frequency that a given symbol s occurs at the head s_{head} or at the tail s_{tail} of a compound, divided by the overall frequency of the symbol:

$$\mathrm{P}_{\text{head}}(s) = \frac{\mathrm{Count}(s_{\text{head}})}{\mathrm{Count}(s)} \tag{4.4}$$

$$\mathrm{P}_{\text{tail}}(s) = \frac{\mathrm{Count}(s_{\text{tail}})}{\mathrm{Count}(s)} \tag{4.5}$$

If a symbol occurs mainly as compound head its head probability will be close to one: $\mathrm{P}_{\text{head}}(s) \approx 1$. If a symbol is always a compound constituent, the sum of its head and tail probabilities will be one: $\mathrm{P}_{\text{head}}(s) + \mathrm{P}_{\text{tail}}(s) = 1$. This is the case for *bounded* constituents, as introduced in Section 4.5.3, as well as most constituents containing a linking morph.

First, we compute P_{head} and P_{tail} for all symbols in the vocabulary using our training corpus. Afterwards, word graphs are generated using the constituent-based system introduced in Section 4.5.1. We then perform a search in the word graphs. There, two symbols s_1 and s_2 are rejoined to a compound word if they occur in the list of in-vocabulary compounds, or alternatively, if their compound probability $\mathrm{P}_{\text{head}}(s_1)\mathrm{P}_{\text{tail}}(s_2)$ is above a given threshold. Since, as mentioned before, this search algorithm works incrementally, considering two constituents at a time, this probability is relevant only for two-constituent compounds. If two constituents are rejoined to a compound word, this compound can in turn become part of a longer compound.

Preliminary experiments showed that many compounds with a large number of constituents were generated. This contradicts what we found in Section 4.4, namely that the vast majority of compounds consist of two or three constituents. Hence, to restrict the number of long compounds generated, the percentages shown in Table 4.4 are used as *a-priori* probabilities λ_N, which are multiplied by the compound probability of every compound candidate. In practice, this additional constraint means that only compound words with two or three constituents ($N = 2$ or $N = 3$) are hypothesized.

In summary, a sequence of constituents c_n is considered to be a compound word if it fulfills the following criterion:

$$\lambda_N \, \mathrm{P}_{\text{head}}(c_1) \prod_{n=2}^{N} \mathrm{P}_{\text{tail}}(c_n) \quad \geq \quad \theta \tag{4.6}$$

Where θ is a threshold used to gauge the number of compound words generated.

The results are disappointing, as shown in Table 4.7. On the one hand, when the constraints imposed to θ are relaxed, a larger number of OOV compounds can be recovered. On the other hand, the overall recognition accuracy also decreases. Obviously, the algorithm tends to join compound words at wrong places in the word graph. This, in turn, suggests that head and tail probabilities do not sufficiently discriminate between non-compound words and compound constituents.

System	WA (all words)	Recall (all OOV)	Precision (all OOV)	Recall (compounds)
Rejoined constituents	60.7 %	[17]1.5 %	17.4 %	0 %
$\theta = 0.50$	59.3 %	17.1 %	17.0 %	14.0 %
$\theta = 0.40$	57.0 %	28.1 %	15.3 %	23.2 %
$\theta = 0.35$	55.8 %	30.8 %	14.0 %	30.2 %

Table 4.7: Performance comparison between the rejoined-constituent system introduced previously and the system aimed at recovering OOV compounds. For the latter, different thresholds θ, as given by Equation 4.6 are tested. We report word accuracy (WA), recall and precision of all OOV words, and recall of compound OOV words.

One reason for this negative outcome may be the lack of training material. Many constituents appear only a few times in our training set and tend to have high head and tail probabilities. The prediction power of such infrequent constituents is, however, much less than frequent ones. In order to compensate for this, we use a simple logarithmic model of reliability: The reliability of a symbol s grows with the log of its frequency $\log \mathrm{Count}(s)$. This measure was incorporated into the algorithm as an additional penalty. The condition given by Equation 4.6 now reads:

$$\lambda_N \, \mathrm{P}_{\mathrm{head}}(c_1) \prod_{n=2}^{N} \mathrm{P}_{\mathrm{tail}}(c_n) \prod_{n=1}^{N} \left(1 - \frac{1}{1 + \log \mathrm{Count}(c_n)}\right) \quad \geq \quad \theta \tag{4.7}$$

Table 4.8 shows that this additional constraint does provide small improvements against the results presented in Table 4.7. But still, the system based exclusively on a list of permitted compound words cannot be outperformed. We conclude that head and tail probabilities are not qualified for recovering OOV compounds, especially in our medium-sized corpus. We therefore abandon this approach at this point.

[17]In this system, a primitive garbage model was used to cover word fragments. Therefore, the recall of the rejoined-constituent system is sightly above zero.

System	WA (all words)	Recall (all OOV)	Precision (all OOV)	Recall (compounds)
Rejoined constituents	60.7 %	[17]1.5 %	17.4 %	0 %
$\theta = 0.50$	59.4 %	17.9 %	17.8 %	11.6 %
$\theta = 0.40$	57.7 %	28.9 %	16.7 %	23.2 %
$\theta = 0.35$	56.8 %	31.6 %	15.8 %	27.9 %

Table 4.8: Performance comparison between the rejoined-constituent system introduced previously and a system aimed at recovering OOV compounds. For the latter, different thresholds θ, as given by Equation 4.7, were tested. Word accuracy (WA), recall and precision of all OOV words, and recall of compound OOV words are shown.

4.6.2 Compound Class Probability

The bad results obtained in the last section are partially due to the sparseness of our training set. In order to robustly estimate head and tail probabilities for a compound constituent, many samples of this constituent are necessary. Unfortunately, though, the majority of the constituents occur only a few times in the training material. In an attempt to overcome this limitation, we will now test an approach based on syntactic classes of constituents. This approach has also linguistic foundations. As explained in Section 4.4, compounding is constrained by certain syntactic rules. For example noun-noun compounds are very common in German, whereas verb-verb compounds are very rare.

In order to quantify this knowledge, we computed what we dubbed *compound class probability*. The compound class probability is the conditional probability that the underlying symbols of a given class pair $\mathcal{C}_l, \mathcal{C}_r$ do in fact form a compound word.

$$\mathrm{P}(w \text{ is a compound} \mid \mathcal{C}_l, \mathcal{C}_r) = \frac{\mathrm{Count}(\mathcal{C}_l, \mathcal{C}_r, w \text{ is a compound})}{\mathrm{Count}(\mathcal{C}_l, \mathcal{C}_r)} \tag{4.8}$$

The compound class probability was calculated for all class pairs in the corpus. As can be seen from Figure 4.15, only a few class pairs are likely to form a compound word. For this reason, we foster the hope that the compound class probability will provide better discrimination capabilities than the head and tail probabilities presented in the last section.

Experiments were conducted using the same method as in the last section: A sequence of symbols (words or constituents) is rejoined to a compound word if its

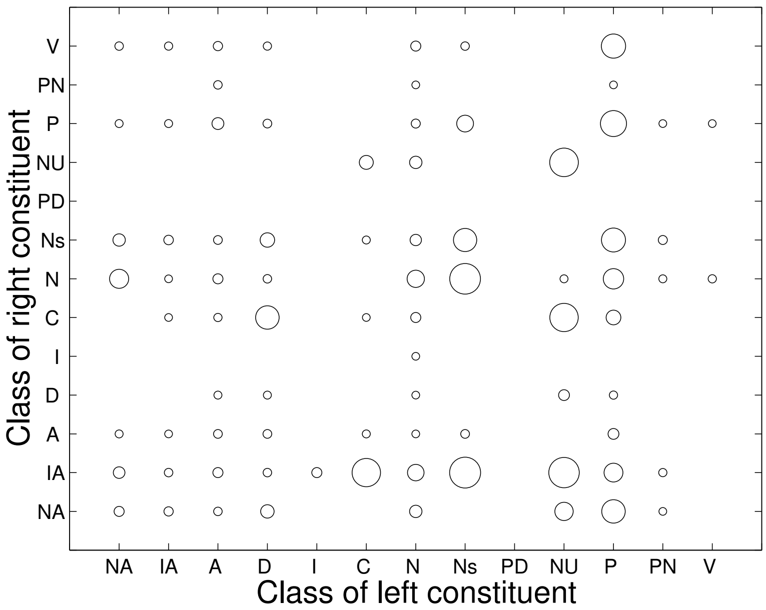


Figure 4.15: Compound class probabilities for all class pairs. The classes of "left" constituents are shown on the abscissa, and the classes of "right" constituents on the ordinate. The radius of the circle at the intersection of class $\mathcal{C}_l$ and class $\mathcal{C}_r$ is proportional to the probability $\mathrm{P}(w \text{ is a compound}) \mid \mathcal{C}_l, \mathcal{C}_r)$.

compound class probability is above a given threshold:

$$\mathrm{P}(w \text{ is a compound}) \mid \mathcal{C}_l, \mathcal{C}_r) > \theta \tag{4.9}$$

Table 4.9 summarizes these results. It can be seen that as the constraints used in the algorithm are loosened, the recall rate of compound words increases. At the same time, however, the word accuracy decreases. Still, the deterioration is not as important as in previous experiments. The class-based approach in fact seems to partially overcome the lack of training material.

4.6.3 Summary of Results

Figure 4.16 compares the various results obtained in this section. The upper bar graph shows the word accuracy decrease (or gain) relative to the rejoined-constituent system. The lower bar graph shows the recall rate of OOV compound

System	WA (all words)	Recall (all OOV)	Precision (all OOV)	Recall (compounds)
Rejoined constituents	60.7 %	[17]1.5 %	17.4 %	0 %
$\theta = 0.25$	60.8 %	3.8 %	31.3 %	2.3 %
$\theta = 0.30$	59.6 %	27.0 %	18.9 %	23.3 %

Table 4.9: Performance comparison between the rejoined-constituent system introduced previously and the system based on class compound probabilities. For the latter, different combinations of the threshold θ, as given by Equation 4.9, and the compound word penalty σ were tested. Word accuracy (WA), recall and precision of all OOV words, and recall of compound OOV words are shown.

words. In the first experiment (on the left) head and tail probabilities were used to decide whether the candidate considered is in fact a compound word. In this approach, a high price in terms of word accuracy deterioration must be paid for recovering OOV compounds. We attempted to improve these results incorporating a reliability metric. The bar graph in the center of Figure 4.16 shows that small improvements were indeed obtained, but the performance deterioration is still intolerable.

We attempted to overcome the lack of training material of the previous approach by incorporating knowledge about the word classes involved in word compounding. Using compound class probabilities we obtained the best results so far (shown on the right side of Figure 4.16): roughly one fourth of all OOV compounds could be recovered at the expenses of 1 % decrease in the word accuracy of the system. Since we are not willing to compromise the overall system performance, we view this outcome as negative. Finally, experiments with a hybrid system, which combines head and tail probabilities with class information were conducted. Unfortunately, though, this attempt provided no improvements.

Our conclusion is that the corpus statistic metrics investigated are not adequate as a generative model of compound words. It appears to us that such a model can only perform satisfactorily if it incorporates higher knowledge sources such as semantic or domain knowledge.

4.7 Summary

In this chapter it was shown that a considerable portion of all German words is affected by some morphological mechanism, and also that OOV compound words

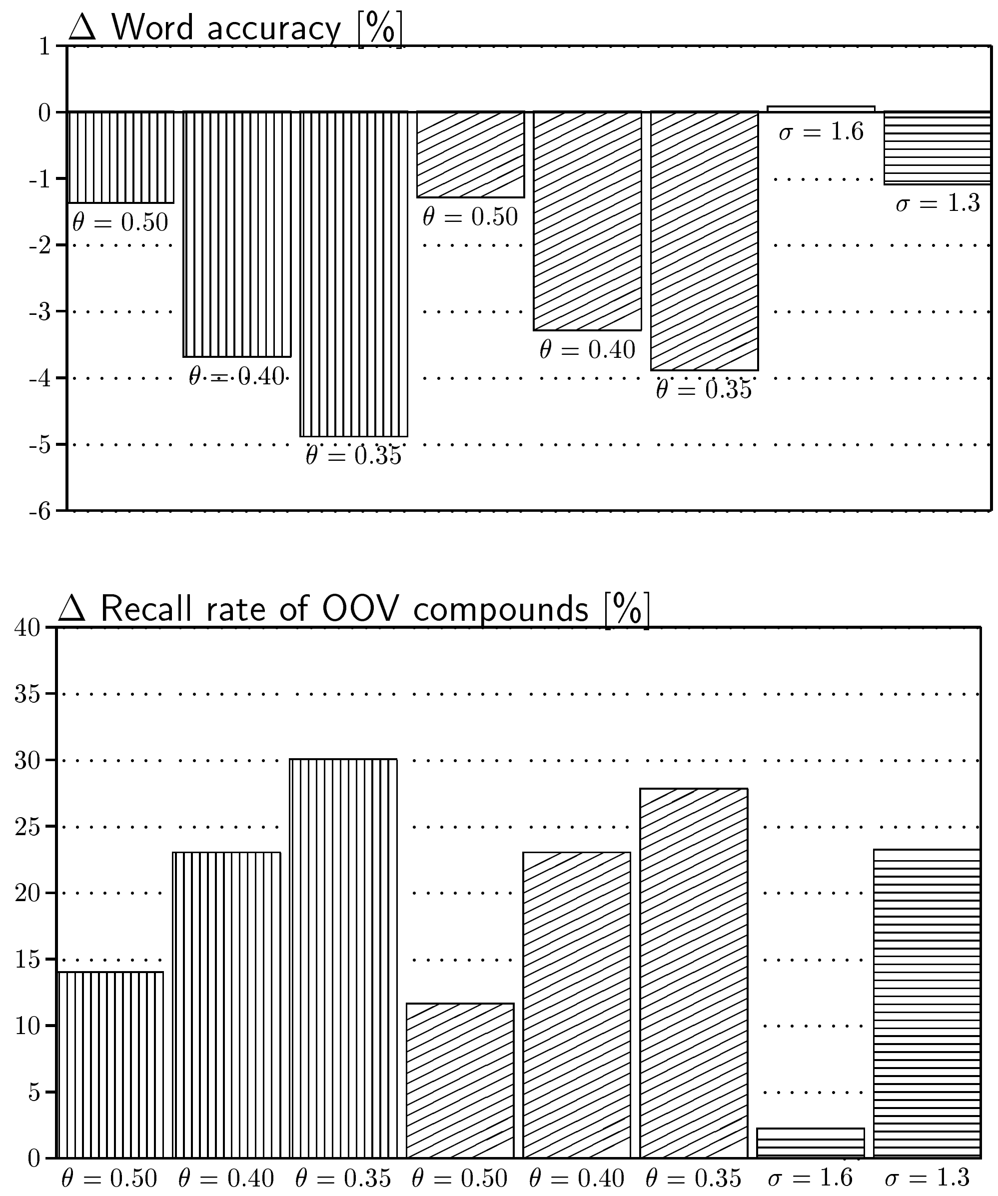


Figure 4.16: Comparison of the various results obtained in Section 4.6. The upper bar graph depicts the word accuracy relative to the rejoined-constituent system. The lower bar graph shows the recovery rate of OOV compounds. The bars on the left refer to the system which uses head and tail probabilities. In the center, a reliability model was also included. On the right side, the results obtained with compound class probabilities are shown.

are mostly made up of in-vocabulary components. These OOV words could be "recovered" by reassembling them from their component parts.

In order to achieve this, morphological information was incorporated into the recognition process. Automatically derived morphologically-oriented lexical units were used to reduce the vocabulary size and improve the recognition performance. Furthermore, various statistics on the composition of compound words were collected from the training set and used to rejoin in-vocabulary components to OOV compounds. Unfortunately, this approach did not yield satisfactory results.

5 DETECTING OOV WORDS

IN this chapter we will concentrate on the problem of detecting out-of-vocabulary words. Various techniques for explicitly modeling OOV words at the acoustic and language modeling stages of our speech recognizer are presented and evaluated. In addition, an error analysis will show the influence of OOV words on the performance of the system.

5.1 Introduction

Given the great significance of out-of-vocabulary words in terms of frequency and semantic importance shown in Chapter 3, it is desirable that whenever speech recognition systems encounter OOV words, they react in appropriate ways. Unfortunately, though, the standard response of a recognizer is to (silently) misrecognize the OOV word without issuing any warning to the user.

In case the user has direct access to the recognizer hypotheses (for instance, in dictation systems), out-of-vocabulary errors can easily be identified and corrected with little more effort than any other in-vocabulary error (by inputting the OOV word using a keyboard, for instance). Ultimately, the new word can be added to the system lexicon, as will be discussed in Chapter 6.

On the other hand, if the speech recognizer is a transparent module of a larger application, for example the VERBMOBIL speech-to-speech translation system, the situation is more difficult. Whereas in-vocabulary recognition errors can often be repaired (by repeating the utterance and speaking more clearly, for instance) [McNair and Waibel, 1994], out-of-vocabulary errors cannot. The problem is that the user cannot differentiate between in-vocabulary and out-of-vocabulary errors because they do not have access to the recognizer hypotheses. Moreover, since most naive users are likely to ignore the vocabulary limitations of the system, they will probably assume that all errors are in-vocabulary errors, and behave accordingly (repeating the same utterance more clearly, for instance). Since recognizers usually have no chance to identify OOV words, this leads to very frustrating experiences for the users. If, on the other hand, the system were in the position to detect OOV words, it could warn users about its vocabulary limitations, and avoid the situation described above [Boros et al., 1997].

In this chapter we will investigate methods aimed at detecting OOV words. After reviewing research reported in the literature in Section 5.2 we will concentrate on augmenting the models used in the recognizer so that they can deal with OOV words. In Section 5.3, a simple and effective scheme to cover the acoustic signal of generic OOV words is introduced.

Acoustic models alone, however, do not take advantage of contextual information. It is intuitively clear that it is much more probable for an OOV word to follow, say, the sequence *"my name is ..."* than, say, *"let's meet on ...."* In order to account for this, in Section 5.4 we concentrate on explicitly modeling OOV words in the language model. There, we consider two typical situations: First, when OOV words are present in the training material of the language model; and second when they are not.

If large amounts of training material are available, the vocabulary can be created using only part of the data, or chosen to contain only the most frequent words. In any case, OOV words are observed in their natural contexts, and this information can be directly captured in the language model. This scenario will be examined in Section 5.4.1.

In the case of speech recognition tasks dealing with spontaneous speech, on the other hand, the little data usually available is frequently entirely used for training. The question that arises is: How can the context of OOV words be modeled when *all* words in the training corpus are in the system vocabulary? In Section 5.4.2 the method of *iterative substitution* for correcting distortions caused by OOV words in the language model will be introduced. It will be shown experimentally that a considerable portion of OOV words can be detected using iterative substitution.

5.2 Prior Work

Next we will review previous publications dealing with acoustic and language modeling of OOV words, as well as using confidence measures towards detecting OOV words. In addition, since garbage modeling in keyword spotting is a topic closely related to OOV word modeling, we will discuss it briefly in Section 5.2.3.

5.2.1 Acoustic Modeling

Asadi was the first to identify the importance of OOV words in speech recognition systems. He and his colleagues presented a series of papers aimed at first detecting OOV words [Asadi et al., 1990], then transcribing them [Asadi et al., 1991], and finally adapting the system to new words which have been incorporated into the recognizer after the original training [Asadi and Leung, 1993].

In [Asadi et al., 1990] four explicit models for OOV words were presented and evaluated:

- a concatenation of five "flat" (untrained) HMMs,
- two different topologies of fully interconnected context-independent phoneme models which require at least two or four phonemes to be observed, and
- a model consisting of fully interconnected diphones.

Experiments were performed with the DARPA 1000-Word Resource Management corpus of American English. OOV words were simulated by deleting entries in some open word categories (e.g. ship names). In the grammar used during recognition, additional arcs were inserted in the classes where the occurrence of OOV words was permitted. The probability of these arcs was reduced by an experimentally optimized factor, in order to reduce the number of false alarms. The best model (fully interconnected context-independent phoneme models requiring at least two phonemes) was able to detect almost three fourths of all OOV words in the test set at a false alarm rate of only 3.4 % of all sentences. These results are extremely positive at first glance, but their relevance is reduced by two characteristics of the experimental framework used: First, the grammar allows OOV words only in particular "slots" of the utterance, and second, all experiments are speaker dependent.

In [Asadi et al., 1991] also speaker-independent experiments were presented. Using the same OOV-word model as before, the detection performance decreased from 71 % to 20 % (1 % of the sentences contained false alarms). When 2 % false alarms were allowed, the detection rate was 50 %. This result is very positive for the application being considered. In my opinion, though, it is still too optimistic for more unconstrained applications. Unfortunately, no comments on the computational requirements of this approach are made in the papers. Considering the unconstrained structure of the generic model, the complexity added to the search is supposedly very high (see discussion on the paper by Sakamoto and Matsunaga [Sakamoto and Matsunaga, 1995] below).

In [Kita et al., 1991], Kita et al. presented a detection approach for OOV words based on two competing grammars: First, the *task grammar*, which encodes information on word sequences allowed; and second, the *phonetic grammar*, which describes the phonological structure of Japanese. The grammars are merged, and act as a finite state machine that directs an HMM-based search. The recognition results can be either in-vocabulary words encoded in the task grammar, or the phonetic transcription of OOV words, as given by the phonetic grammar.

Experiments were carried out with an extremely small corpus (which contained only 14 phrases). Therefore, we cannot draw general conclusions from the results presented, which seem to be discouraging: Compared to the system that uses only a task grammar (thus allowing only in-vocabulary words) the performance of the system that also includes a phonetic grammar decreases considerably. Furthermore, the computational cost involved in the additional phoneme recognition needed to detect OOV words is very high. This is also true for other works based on the same principle and reviewed below.

Itou et al. presented a similar approach in [Itou et al., 1992]. The speech recognizer was also controlled by an LR parser for Japanese, which determined the allowed word sequences, and they also used a second grammar, a phoneme trigram, to model OOV utterances. OOV words are identified by thresholding the difference between the scores provided by these two grammars.

The test database consisted of simple sentences generated by a grammar. For testing, some words were deleted from the vocabulary to simulate the occurrence of OOV words. This produced a very high detection rate of 75 % (at the expense of a high false alarm rate of 11 % at the sentence level). The very positive result obtained with this simple approach suggests that the task used for testing is rather simple.

In the continuation of this work, the authors again used two recognition processes working in parallel, and compared the acoustic scores of a phonetic recognizer with a lexically constrained recognizer [Hayamizu et al., 1993]. Whenever the difference between the scores was higher than a given threshold, an OOV word was hypothesized.

Experiments were conducted on a Japanese isolated-word task with only a single speaker being tested in speaker-dependent mode. Different vocabulary sizes were used, and in all cases the word being tested was deleted from the recognition lexicon. As expected, the results of the OOV tests were significantly worse than when the test word was included in the vocabulary. An error analysis showed that, for smaller vocabularies, false alarms make up a major portion of all errors. As the vocabulary size increases, however, the number of undetected OOV words grows and becomes the main error source. This occurs because the score distributions of in-vocabulary and OOV words are highly overlapped, thus making a thresholding approach ineffective. This problem becomes worse as vocabularies grow larger and word confusability increases. Even though this approach is appealing for its intuitiveness, the experimental results presented are rather disappointing, even under the simplified testing conditions used.

In [Yamada et al., 1994] OOV-word detection is used to dynamically switch among different recognition vocabularies: Whenever an OOV word of a particular class is detected, the vocabulary of that class is expanded using supplementary dialog information, and then the unknown segment is re-recognized. For detection, a simple mixture of two HMMs is used, one trained on all vowels and the other on all consonants. This explicit model competes with in-vocabulary models during recognition, and is hypothesized just as any other word. The main difference to the other research discussed above is that the OOV-word model used here is much simpler and less specialized, therefore requiring much less additional computation during recognition.

Experiments were performed using a small-vocabulary travel information system for Japanese. Recognition was guided by a grammar that included additional arcs where proper names were allowed. Under these constraints, an almost perfect detection rate was reported (94 %). The overall performance of the system, however, was considerably lower than when testing under closed vocabulary condition, which suggests that the high detection rate was achieved at the cost of a high false alarm rate.

Sakamoto and Matsunaga [Sakamoto and Matsunaga, 1995] also used less specialized models consisting of either one cluster model (trained with all phonemes); or one consonant cluster (trained with all consonants) connected in parallel[1] with separate models for each Japanese vowel. The performance of these models was compared with a model including all phonemes in parallel[1] (as in Asadi's work). Again, recognition was guided by a grammar (a context-free grammar), in which OOV words are permitted in nouns slots. Under these conditions, the authors argue that the model including one consonant cluster provides the best detection capabilities and require less computation than the more specific all-phonemes model.

Experiments with this model were conducted on a set of six speakers in a Japanese conference registration task. When the test was run with a closed vocabulary (that includes all words spoken in the test), the word accuracy was 72 %. With an open vocabulary the performance decreases significantly to 40 % without OOV-word modeling, or to 54 % with OOV-word modeling. In the latter system, the computational requirements are reported to grow by a factor of two with respect to the closed vocabulary system.

Most of the research reviewed above introduced or refined a technique aimed at detecting OOV words in the framework of a recognizer guided by some kind of deterministic grammar (CFG, LRG, etc.). In this context, OOV words are only permitted in predetermined "slots," and consequently the *a-priori* probability of OOV words in these slots tend to be high. Therefore, the experimental results

[1] The transition probabilities of this model were provided by a phoneme 4-gram.

presented are only valid within these very constrained testing conditions used. When using a stochastic language model, OOV words can occur virtually everywhere, thus making OOV-word detection much more difficult. This will be the scenario in the papers reviewed next (as well as in our own work presented below). Note the dramatic decrease in the detection rates reported below with respect to the previous studies.

In [Jusek et al., 1995] two generic German syllable models were built using phonotactic knowledge. The syllables are represented by a phoneme network which encodes allowed transitions between consonants or consonant clusters, and vowels or diphtongs. The OOV-word model proposed consists of a concatenation of phonotactic syllable networks. This model can be viewed as a more constrained Asadi model, which does not allow arbitrary phoneme sequences, but only those valid in German. As in Asadi's work, the OOV-word model was incorporated in the recognition lexicon, context independent phoneme units were used at the network transitions, and a bias was introduced to fine-tune the scores. Also, due to the large number of possible paths in the phonotactic network, the computational costs involved in this method are very high.

Experiments were performed on the 1995 Evaluation set of the VERBMOBIL corpus—the same test set used in Chapter 4 of this thesis. The OOV-word model proposed was able to detect roughly one fourth of all OOV words. Unfortunately, though, the overall accuracy of the system decreased slightly compared to the system without OOV-word modeling.

An extension of this approach was presented in [Kemp and Jusek, 1996]. Here, the phonotactic network was augmented with transition probabilities. These transition probabilities were computed by counting the transitions involved in the phonotactic network when traversing it with the phonetic transcriptions of a given set of words. In addition, a data-driven syllable model was automatically derived from the training corpus. In order to achieve this, a minimal graph that encodes all syllables occurring in the training corpus was computed. Consequently, no phonetic knowledge other than the phonetic transcriptions of all words was necessary to create this model. The transitions in this graph were also weighted using the counts derived from the training corpus (as in the "augmented" phonotactic model).

Instead of incorporating the model as a single lexicon entry, the authors propose adding all states involved in the model (which are phonemes) into the lexicon, and register the transitions directly in the language model used during recognition. For this, the standard (word-based) language model in complemented with the phoneme transition probabilities computed from the corpus.

The models were tested on a subset of the VERBMOBIL 1995 evaluation material. The phonotactic network provided this time a small improvement in the overall accuracy of the baseline system, and was able to detect 10 % of all OOV words. When augmenting the phonotactic network with transition probabilities, or when using the data-driven model, the detection rate roughly doubled, but the number of incorrectly hypothesized OOV words tripled. Since the word accuracy remained constant, the false hypotheses seem to occur mainly at places where the recognizer made an error anyway. This is an undesired behavior if OOV-word hypotheses are to be processed by higher-level knowledge sources (such as parsers).

5.2.2 Language Modeling

Most of the studies reviewed in the previous section used very constrained testing conditions, with grammars that allow OOV words to occur only in particular slots. This way, a very important aspect of OOV-words, namely modeling their context, was ignored, and the task of detecting OOV words was artificially simplified. When dealing with more unconstrained tasks, detecting OOV words becomes considerably more difficult, and a powerful language model that contributes to OOV-word detection is indispensable.

Suhm et al. were the first to identify this, and showed the importance of a language model which is especially tuned to OOV word detection [Suhm et al., 1993]. For the acoustic models, they used an automaton which allowed an arbitrary sequence of phonemes, similar to the new word model proposed by Asadi et al. in [Asadi et al., 1990]. For the language model they mapped all words in the training corpus which were not included in the vocabulary into one symbol, in effect explicitly modeling the context of a generic OOV word.

This approach was tested on a small-vocabulary read-speech corpus in a conference registration domain. All proper names were considered OOV words, and excluded from the vocabulary. When using the modeling scheme described above, a (very high) detection rate of approximately 70 % was achieved (with a false alarm rate of 2 % of all spoken words). This also drastically improved the overall accuracy of the system with no OOV-word modeling from 71.7 % to 78.6 %. The higher accuracy of the system that also includes proper names in its vocabulary (88.5 %) shows that there is still plenty of room for improvement.

In [Fetter et al., 1995] we introduced the method dubbed *iterative substitution*, which compensates distortions produced in the language model due to lack of training material. Particularly, if all words seen so far are used for language model training, the probability mass allocated to OOV words will be artificially small. Iterative substitution simulates the occurrence of OOV words, contributing to a better modeling of its context. A more detailed discussion of this method will be presented in Section 5.4.2.

Gallwitz et al. extended our method for class-based language models [Gallwitz et al., 1996]. Instead of estimating a single probability of observing an OOV word in the test material, they computed one OOV-word probability for each word class in their language model. On the acoustic side, a single model with equally distributed density functions for all OOV-word classes was used. In this method, when an OOV word is detected, its class is also available, which can be advantageous for higher-level processing modules.

Tests were performed in the framework of the EVAR train timetable inquiry system (see Section 3.3.1). Using this approach, 15 % of all OOV words were detected and the word error rate of the baseline system was reduced from 22.9 % to 21.5 %. This is a significant improvement, particularly considering the high false alarm rate of 75 % reported[2]. Apparently, OOV words are mainly incorrectly hypothesized where other recognition errors which would have occurred anyway.

In the continuation of this work, the linguistic processor and the dialog manager of the EVAR system were extended to properly handle OOV word hypotheses [Boros et al., 1997]. The detection capabilities of the method were similar to those reported before (24 % recall and 31 % precision). In addition, the classification accuracy was found to be very high: Whenever an OOV word was correctly detected, the hypothesized word class was correct 94 % of the time.

5.2.3 Keyword Spotting

Out-of-vocabulary-word modeling in continuous speech recognition and garbage modeling in keyword spotting are related techniques. In both cases the duality between vocabulary words and OOV words is present, but vocabulary sizes (and consequently the magnitude of the OOV problem), and the relative importance of OOV word detection are different.

Wilpon et al. pioneered the application of HMM-based systems for keyword spotting. In [Wilpon et al., 1990] they experiment with garbage models, which are explicitly trained models of noises and extraneous (out-of-vocabulary) speech input. Different scenarios for three different types of label information available for the acoustic training of the garbage models were tested:

1. Keywords and *individual* non-keywords are labeled in the training database.

2. Keywords and non-keywords (as a single class) are labeled in the training database.

3. Keywords are only labeled in the training database.

[2]A high false alarm rate usually has a negative effect on the word accuracy.

The results are surprising: Using a large number of specific garbage models does not necessarily improve recognition rates of keywords embedded in noises or extraneous speech. In fact, the results obtained using a training database where only keywords are labeled (case 3) and a fully automatic training procedure to generate a single garbage model are comparable to those obtained using a labeled database (case 1) for training many specific garbage models.

In [Boite et al., 1993] a new approach for modeling OOV words in keyword spotting was presented. Garbage-model probabilities are not explicitly trained *(off-line)*, but estimated *on-line* as an average of the (emission) probabilities of the models used for keywords. This method has the advantage of not requiring extra training for the garbage models. Furthermore, it can be applied independently of the task vocabulary, provided that models for the vocabulary exist.

The paper presented a comparative study to evaluate explicit and on-line garbage models. As usual in keyword spotting, utterances are constrained to a syntax consisting of a leading succession of zero or more garbage models, followed by one or zero keywords, followed by trailing garbage models. Special attention is given to the transition probabilities between garbage models and keywords, and between garbage models, which are called *penalties*. These transition penalties are absolutely necessary (especially when using on-line garbage models), since the garbage HMM scores per frame are for the most part lower than those of the keywords competing in the search. The penalties improve recognition results, and on-line garbage models exhibit a similar performance to explicit models.

A major research issue in the field of keyword spotting is the adaptation of Large Vocabulary Continuous Speech Recognition (LVCSR) systems to the particularities of keyword spotting tasks. One of the first efforts in this direction was presented in [Weintraub, 1993]. There, the keyword likelihoods were estimated using time-normalized Viterbi scores (i.e. frame scores). This approach has the advantage of incorporating language modeling information on the sequence of words containing the keyword. The disadvantage, however, is that it measures the absolute quality of the match between models and data, where a relative quality measure would be more beneficial.

This drawback was handled in the continuation of this work [Weintraub, 1995]. The technique presented was dubbed LVCSR log-likelihood ratio scoring. The proposed scoring parameter is formulated as the ratio between the number of word sequences containing the keyword in question and all possible word sequences (both weighted by the probability of the respective word sequence).

This ratio was implemented using N-best lists[3], that is “all possible word sequences” is approximated by “the words sequences contained among the N-

[3]This is why LVCSR log-likelihood ratio scoring is also called N-best likelihood ratio.

best candidates." Using LVCSR log-likelihood ratio scoring, a significantly better performance was obtained than with time-normalized Viterbi scores.

In [Meliani and O'Shaughnessy, 1996] El Meliani et al. propose using common techniques for both OOV-word modeling in large-vocabulary continuous speech recognition and garbage (or filler) modeling in keyword spotting. They present experimental results on using different kinds of filler models at the lexical level (that is, garbage models are just lexicon entries, and not additional HMMs as in [Wilpon et al., 1990]). The models tested include both single and multiple lexical phonemic fillers and lexical syllable fillers. Unfortunately, though, the reported results are not relevant, because a dramatic decrease in word accuracy is accepted to achieve a higher detection rate.

5.2.4 Confidence Measures

So far we have reviewed several approaches aimed at solving the OOV-word problem by incorporating explicit models into the system[4]. An alternative to these methods consists of modeling OOV words not explicitly but implicitly by looking at certain parameters during recognition. These parameters are usually termed "confidence measures."

Ideally, confidence measures should have a value of 0 when the recognizer makes a mistake, and 1 otherwise. This ideal behavior can in practice be approximated, and one can expect confidence measures to be highly correlated with the probability of the recognizer decisions being correct. A high confidence value would consequently indicate a good match of the input utterance to the system models. On the other hand, confidence would be low whenever the recognizer is forced to match its models with extraneous input.

Having loosely described the requirements confidence measures should fulfill, we will now discuss which recognizer parameters could be used as confidence measure. In the framework of an ideal Bayes' classifier, the best choice for a measure of confidence is the *a-posteriori* probability $\mathrm{P}(\omega \mid \mathbf{v})$ that an observation v belongs to the class ω. In fact, $\mathrm{P}(\omega \mid \mathbf{v})$ *is* the probability that the decision made by the classifier is correct [Schürmann, 1996].

Unfortunately, though, in real speech recognition systems the situation is more complicated. As explained in Chapter 2, speech recognizers can estimate the most likely word sequence for a given spoken utterance. They can further rank a list of

[4]Exceptions are on-line garbage modeling in keyword spotting (Section 5.2.3), and perhaps also the detection approach reviewed in Section 5.2.1 based on an additional phoneme recognizer running in parallel with the word recognizer, which was used by [Kita et al., 1991], [Itou et al., 1992], and [Hayamizu et al., 1993]. These approaches can also be viewed as applications of confidence measures.

candidates, which are defined in the system lexicon. Every candidate is assigned a *score*, which is estimated by combining different knowledge sources using Bayes' rule. These scores are no longer *a posteriori* probabilities, but rankings, whose validity is limited to the scope of the current utterance. This is due to a number of simplifications and heuristics used during the recognition process, including ignoring *a priori* probabilities, using empirically-optimized weighting factors for different knowledge sources, etc. Therefore, recognizer scores are generally not suited as confidence measures.

Due to this fact, a large number of publications devoted to this subject concentrate on finding parameters which show affine characteristics with the desired behavior of confidence measures. A short (and incomplete) selection of these proposed measures of confidence is listed below.

- The "a-posteriori confidence" based on the acoustic scores [Young, 1994], [Fetter et al., 1996a].
- The "a-posteriori confidence" based on normalized acoustic scores [Young, 1994], [Cox and Rose, 1996], [Fetter et al., 1996a].
- Statistics derived from the language model [Eide et al., 1995], [Schaaf and Kemp, 1997].
- Statistics derived from counting parallel hypotheses in a word graph [Eide et al., 1995], [Schaaf and Kemp, 1997].
- The number of alternatives competing in the search [Hetherington, 1995a], [Cox and Rose, 1996], [Schaaf and Kemp, 1997].
- A combination of (some of) the items enumerated above [Cox and Rose, 1996], [Schaaf and Kemp, 1997].

Being able to reliably estimate the confidence of the hypotheses allows the recognizer either to reject all or part of the utterance and prompt the user for clarification, or to revoke the original choices in favor of more confident candidates. Furthermore, confidence measures can also be used to combine knowledge sources, and choose utterance segments for unsupervised adaptation.

Obviously, confidence measures can be very beneficial. But how relevant are they to the topic of this thesis? As discussed above, confidence measures are not exclusively aimed at detecting OOV words, but portions of the input signal that do not match the existing models well. This can happen, for example, when the input utterance is unclearly pronounced, very noisy, or in fact does contain OOV words. Due to this lack of specificity, the effectiveness of confidence measures as

"OOV-word detectors" highly depends on the percentage of errors caused by OOV words in the task in question. If a major portion of the errors are due to OOV words, confidence measures can provide very good detection capabilities. If not, confidence measures will not be good OOV-word detectors. Since the hypotheses of such a detector will mostly be errors of other kinds, its precision (as defined in Equation 2.24) will be correspondingly low.

The VERBMOBIL corpus used for recognition experiments in this thesis, belongs to the latter category. Whereas the error rate (achieved by our recognizer) in VERBMOBIL is over 20 %, the OOV rate is less than 2 % (see Section 5.4.2). That is, OOV words contribute directly to less than 10 % of all errors[5]. Therefore, we believe that using confidence measures is not an effective strategy in detecting OOV words in VERBMOBIL (at the performance level mentioned above).

This was also confirmed experimentally. In [Fetter et al., 1996a] we showed that the system performance can be significantly improved using the information provided by confidence measures as an additional knowledge source during recognition. But we were unable to significantly improve the detection rate of OOV words. In the continuation of this work (not yet published), we tested additional parameters and, again, confirmed this conclusion. Therefore, this topic will not be further pursued in this thesis.

5.3 Acoustic Modeling

The OOV-word detection approaches reviewed in the last section can be classified in two broad groups: explicit and implicit modeling schemes. *Explicit* OOV-word models actively compete against models of in-vocabulary words during the word-based search. They were discussed in parts of Section 5.2.1 and Section 5.2.2 (acoustic and language modeling of OOV words, respectively), and also when reviewing off-line garbage models in keyword spotting (Section 5.2.3). *Implicit* models use information derived from other recognition parameters to compute the likelihood of OOV words. They were discussed in Section 5.2.4 on confidence measures, and in the part of Section 5.2.3 which deals with on-line garbage modeling. In addition, the detection approach based on an additional phoneme recognizer running in parallel to the word recognizer used in [Kita et al., 1991], [Itou et al., 1992], and [Hayamizu et al., 1993], can also be assigned to this category.

Explicit modeling can in turn be differentiated into two approaches: phoneme grammars, and garbage models. In the first approach a grammar is defined which

[5] Mainly due to the contextual information provided by the language model, OOV words produce on the average more than one error. This will be discussed in detail in Section 5.4.1. For the purpose of the current discussion, though, we do not consider this "domino" effect.

allows all possible combinations of phonemes [Asadi et al., 1990], [Suhm et al., 1993], or only some of them [Jusek et al., 1995], [Kemp and Jusek, 1996]. Since the sub-word units are the same for both OOV and in-vocabulary words, no additional training is necessary in this approach. In practice, however, such grammars tend to over-generate OOV-word hypotheses. In order to avoid a large number of false alarms, the scores assigned to the OOV-word models must be penalized by a factor which must be optimized by cross validation. Furthermore, since such grammars have to be very complex to provide accurate modeling, the additional computational requirements are very high. For this reason, this approach has been restricted to research systems, where processing time does not play a big role.

The second approach consists in using additional, dedicated garbage models for OOV words, as in keyword spotting. As discussed in Section 5.2.3, garbage models can be defined at the lexical level or at the word (or model) level. Since lexical garbage models are inadequate for OOV-word detection in large-vocabulary continuous speech recognition, we will consider only word-level garbage models. These models are usually more general and less "sharp" than those in the grammar approach, and a separate training is required. On the other hand, this method does not require any threshold tuning and is easily integrated into an existing system, without any adaptation. The most important advantage of this method, however, is that it requires only very little additional computation during the search. For this reason we chose it for all succeeding experiments.

Since we will use word-level garbage models for OOV words, we decided to train not just one but several models, each specifically aimed at detecting OOV words of a particular length[6]. The system was tested with different numbers of models on a cross validation set, and the best performance was obtained using 15 models (although this number is clearly dependent on the training conditions). Therefore, 15 word-level garbage models for OOV words were adopted for the rest of this chapter. The topology of these models is similar to that of models for frequent words [Class et al., 1993b]: The number of states in each model depends on the length of the word it is aimed to model, as shown in the row labeled "States" in Table 5.1. There are loops for every state and skip arcs for adjacent states.

The models were trained in the standard manner (as described in Chapter 2)

[6]At first glance, this seems to contradict the results obtained in [Wilpon et al., 1990], which show no advantage in using several specific garbage models (see Section 5.2.3, page 96). The different OOV word models used in this thesis, though, are more specific concerning their length only, and not the acoustics they represent. In keyword spotting on the other hand, the length of the garbage models is also controlled (using penalties during recognition). Therefore, both approaches are complementary and not contradictory.

using the complete training material[7]. Every word in the training material was assigned to a particular OOV-word model, according to its length in phonemes. As shown in Table 5.1, all words containing a single phoneme were used to train the OOV-word model number 1. Similarly, OOV-word model number 10 was trained with all words of 10 or 11 phonemes.

As in standard models, ten iterations of the Forward-Backward algorithm were necessary for the probabilities of the models to converge. Cross validation tests showed that the detection capability of the OOV-word models indeed grew steadily in the first iterations, and remained almost constant at the end of the training. This is consistent with Asadi's findings, that untrained OOV-word models provide deficient detection capabilities [Asadi et al., 1990]. Inspection of the emission probabilities of OOV-word models showed that, as expected, they are "flatter" than in both models for polyphones, and more specific whole-word models.

Model	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
States	3	4	5	6	7	8	9	10	11	12	13	14	15	15	15
Length	1	2	3	4	5	6	7	8	9	10–11	12–14	15–18	19–23	24–29	30–

Table 5.1: Characteristics of the acoustic models for OOV words. The number of states of each model, as well as the length of the words (in phonemes) used to train them is shown.

These models provide one of the probabilities needed during recognition, namely the conditional probability of the acoustic signal, as explained in Chapter 2. The other probability, the *a-priori* probability of the word string is provided by the language model, and will be discussed in detail in the next section.

5.4 Language Modeling

Even though the techniques presented in this section do not assume any particular kind of language models, we will use back-off language models for the purposes of explanation. They are the most widely used language models in speech recognition, and were adopted for all experiments in this thesis. In the framework of back-off language models, the bigram probability is given by:

$$\mathrm{P}(w_i \mid w_{i-1}) = \begin{cases} d_r\, \mathrm{P}_{\mathrm{MLE}}(w_i \mid w_{i-1}) & \text{if Count}(w_{i-1}, w_i) > 0 \\ \alpha_{w_{i-1}} \mathrm{P}(w_i) & \text{otherwise} \end{cases} \tag{5.1}$$

[7] When using the entire training material to train OOV-word models, we make the assumption that in-vocabulary words of a particular length have similar acoustic characteristics as OOV words of the same length.

where d_r is the Good-Touring discount coefficient for the count r (see [Katz, 1987]), P_{MLE} the *Maximum Likelihood Estimation* of $P(w_i \mid w_{i-1})$, as given by Equation 2.10, and $\alpha_{w_{i-1}}$ the back-off coefficient for the word w_{i-1}, as described in [Katz, 1987].

In this section, two different situations which directly affect the treatment of OOV words in the language model will be considered: first, when OOV words occur in the training material, and then when the whole training material is used to build the vocabulary of the language model, and consequently, no OOV words occur in it.

For all experiments, the "official" training material defined for the VERBMOBIL Evaluation in 1996 [Reinecke, 1996] was used to train both acoustic and language models. Table 5.2 summarizes the sizes of the training material. The tests were also performed on the test set used for that occasion. A few errors in the reference transliterations were manually corrected[8] and the resulting sizes of the test set are shown in Table 5.2

Corpus	Duration	Dialogs	Turns	Words
Training	28 hours	633	12,001	≈ 268 K
Test	40 min	35	343	≈ 6.6 K

Table 5.2: General statistics of the training and test subsets of the VERBMOBIL corpus used for the recognition experiments presented in this chapter.

5.4.1 Training Material Is Available

If OOV words occur in the training material, they can first be mapped into a unique symbol (say `<UNK>`), and then treated the same as any other vocabulary word. In a back-off language model as described above, this means that the probability of a bigram involving OOV words is either the discounted maximum likelihood estimation of the bigram probability in question (if the bigram has been observed during training), or the backed-off unigram probability of the current

[8]The corrections involve mostly so-called "technical interruptions," which are caused when recording was interrupted by the push-to-talk button. By default, all words affected by technical interruptions are mapped to the symbol `<UNK>` (see Table 3.2 on page 37). We think that this mapping should not be performed in some cases: Whenever the technical interruption affects a valid word (which is already in the vocabulary) this word was left unmodified (and the marker of technical interruptions deleted). In addition, all technical interruptions involving very short speech fragments (with one or two phonemes of short duration) were deleted.

word (if the bigram has not been observed):

$$\mathrm{P}(w_i \mid \texttt{OOV}_{i-1}) = \begin{cases} d_r \, \mathrm{P}_{\mathrm{MLE}}(w_i \mid \texttt{OOV}_{i-1}) & \text{if Count}(\texttt{OOV}_{i-1}, w_i) > 0 \\ \alpha_{\texttt{OOV}_{i-1}} \mathrm{P}(w_i) & \text{otherwise} \end{cases} \quad (5.2)$$

$$\mathrm{P}(\texttt{OOV}_i \mid w_{i-1}) = \begin{cases} d_r \, \mathrm{P}_{\mathrm{MLE}}(\texttt{OOV}_i \mid w_{i-1}) & \text{if Count}(w_{i-1}, \texttt{OOV}_i) > 0 \\ \alpha_{w_{i-1}} \mathrm{P}(\texttt{OOV}_i) & \text{otherwise} \end{cases} \quad (5.3)$$

In both cases, the frequencies necessary for the computations are counted in the training corpus, and no special provision for OOV words is made.

In order to evaluate this situation, we used a vocabulary which includes all words in the VERBMOBIL training corpus, except for the proper names, because they are likely to be OOV words in the VERBMOBIL scenario. The resulting vocabulary contained approximately 4,750 distinct words (or a total of approximately 5,400 lexicon entries, including pronunciation variants), and the OOV rate of the test set was 4.11 %.

Experiments were run first without and then with models for OOV words. No measurable difference in the run time of these two systems was observed. Table 5.3 shows the improvement in accuracy obtained when including OOV-word models in the system. Even though only about one third of all OOV words are detected as such (Recall = 32 %), a high precision is achieved. The precision rate is even higher than the word accuracy, which means that OOV word hypotheses are more likely to be correct hypotheses than in-vocabulary words. The third row of Table 5.3 will be explained in the next section.

System	Recall	Precision	WA
No OOV modeling	0 %	—	73.6 %
OOV modeling	32 %	84 %	75.2 %
OOV modeling and IS	37 %	76 %	75.2 %

Table 5.3: Comparison of performance between a system without OOV-word models, and one with OOV-word models. In both cases, since OOV words were available for training the language model, a standard back-off model is used, and no special provision for OOV words is made. The third row shows additional results obtained when using a language model trained with texts which have been preprocessed by the iterative substitution algorithm presented in Section 5.4.2.

5.4.2 Training Material Is Not Available

When the system vocabulary includes all words that occurred in the training material, so that no OOV words can be observed, the bigram probabilities that involve OOV words are estimated as follows:

$$\mathrm{P}(w_i \mid \mathtt{OOV}_{i-1}) = \mathrm{P}(w_i) \tag{5.4}$$
$$\mathrm{P}(\mathtt{OOV}_i \mid w_{i-1}) = \alpha_{w_{i-1}} \mathrm{P}(\mathtt{OOV}_i) \tag{5.5}$$

Equation 5.4 says that the context information provided by OOV words is ignored. Equation 5.5 says the probability of an OOV word depends on its context (as given by the back-off coefficient $\alpha_{w_{i-1}}$), and on the unigram probability of OOV words. But since no OOV were observed, the maximum likelihood estimate of $\mathrm{P}(\mathtt{OOV}_i)$ is zero. To avoid this, Good-Touring discounting is usually applied (see Section 2.1.5).

The Good-Touring estimate for the probability mass assigned to OOV words is n_1/n_{tot}, where n_1 is the number of words occurring only once (singletons), and n_{tot} is the total number of words in the training material. This probability mass is subtracted ("discounted") from the probability mass assigned to infrequent words by the maximum likelihood principle.

The Good-Touring estimate provides a good results in various tasks. In Verbmobil, however the relative frequency of OOV words is underestimated by the Good-Touring formula. For the training and test materials used for the experiments in this chapter, the OOV rate (as perceived by the language model, see below) is 1.62 %, and $n_1/n_{\mathrm{tot}} = 0.72\,\%$. In order to compensate for this, we proposed the method we dubbed *iterative substitution* [Fetter et al., 1995]. This method allows the occurrence of OOV words in the training corpus to be simulated, as will be explained below. Iterative substitution effectively introduces OOV words into the original corpus, so that Equations 5.4 and 5.5 can be replaced by the more specific Equations 5.2 and 5.3.

All experiments presented in this section were conducted with the training and testing conditions summarized in Table 5.2. The vocabulary this time includes all words that occurred in the training material, resulting in approximately 5,300 words (or 6,000 lexicon entries, including pronunciation variants). The OOV rate measured on the reference texts of the test set was 1.96 %. Since the reference transliterations are slightly modified for language modeling (adding markers for sentence boundaries, for example), the OOV rate perceived by the language model is slightly lower, namely 1.62 %.

Iterative Substitution: The Algorithm

Algorithm 1 The Basic Iterative Substitution Algorithm

In each iteration of this algorithm, the complete corpus in question is arbitrarily divided into a known *and an* unknown *section.*
The known *section in each iteration must include the* known *section of the preceding iteration, and part of the* unknown *section of the preceding iteration.*
The exact division into known *and* unknown *sections within the constraints just mentioned is a matter of trial and error, as is the number of iterations n.*
In addition to the given corpus, an augmented corpus, *which is initially empty, is created. At the end of the n iterations the augmented corpus contains n modified copies of the original corpus.*

1: **LOOP** through Step 6 until the n iterations are complete
2: Make next division of original corpus into *known* and *unknown* sections.
3: Set up a vocabulary of all words in the *known* section of the corpus.
4: Replace all words in the *unknown* section which do not occur in the *known* vocabulary with the symbol `<UNK>`.
5: Append the *known* corpus and the modified *unknown* corpus to the augmented corpus. These two corpora together form a modified version of the original corpus.
6: **END LOOP**

The augmented corpus now effectively contains n modified copies of the original corpus, where n is the number of iterations chosen.

7: Set up a list of all N-grams occurring in this final augmented corpus and how often each N-gram occurs.
8: Divide the N-gram counts by the number of iterations n.

The idea behind iterative substitution is to simulate the process of collecting new data, which will include new, out-of-vocabulary words. A similar method was used in Section 3.4 to analyze the frequency of occurrence of OOV words. Here, the corpus in question is arbitrarily divided into known and unknown sections. Then the unknown section is included piece by piece into the known section, until it is all used up. The divisions into known and unknown sections must be given as parameters when running the algorithm, and are used to control the relative frequency of induced OOV words. Algorithm 1 specifies the precise formulation of iterative substitution.

After all iterations have been performed (after Step 6), the augmented corpus is n times larger than the original one (where n is the number of iterations performed). The counts of all N-grams not containing an OOV word are the same as in the original corpus. If an N-gram does contain an OOV word in the augmented corpus, then its count will be subtracted from the counts of all N-grams in the original corpus which are identical to this "`<UNK>`" N-gram, except for the `<UNK>` positions. Table 5.4 illustrates this.

3-gram *I am ...*	Count (original corpus)	Count (augmented corpus)
I am Pablo	5	2
I am David	4	3
I am `<UNK>`	0	4
Σ Count	9	9

Table 5.4: Example of N-gram counts being modified by iterative substitution.

The effectivity of iterative substitution in improving the detection of OOV words was shown [Fetter et al., 1995] and [Fetter et al., 1996b] (and will be presented below as well). There are, however two problems, namely:

1. The section of the corpus which happened to be defined as *known* at the beginning of the process will not contain any OOV words. The initial sorting of the corpus can therefore have an influence on the results.

2. The partitions of the corpus and the number of iterations are not determined automatically. The optimal values have to be found by trial and error.

In order to solve these problems, we now propose a new modified version of iterative substitution, described in Algorithm 2. The main difference between Algorithm 1 and Algorithm 2 is the partitioning of the original corpus into known and unknown sections, as shown by Figure 5.1.

Even though the modified iterative substitution algorithm (Algorithm 2) has some advantages, experimental results showed no significant differences to Algorithm 1. In order to allow comparisons with our prior work ([Fetter et al., 1995] and [Fetter et al., 1996b]) we therefore preferred to report experimental results obtained with Algorithm 1 (see next Section). The numbers obtained with Algorithm 2 vary only slightly.

Iterative Substitution: Perplexity Evaluation

In order to test the quality of the language model in the vicinity of OOV words, we constrained the definition of perplexity (see Equation 2.19) as follows:

$$\log_2 \mathrm{PP}_{\mathtt{OOV}\rightarrow} = -\frac{1}{n} \sum_{\substack{i=1 \\ w_{i-1} \in \mathtt{OOV}}}^{n} \log_2 \mathrm{P}(w_i \mid w_{i-1}) \qquad (5.6)$$

Algorithm 2 The Modified Iterative Substitution Algorithm

In each iteration of this algorithm, the complete corpus in question is divided into a known *and an* unknown *section.*
The known *section in each iteration does not include the entire* known *section of the preceding iteration, but either a part or nothing of it (as the example of Figure 5.1 shows). The number of words in the* known *section* N_ν*, as well as the number of iterations* n *is calculated as indicated in steps 1 to 4. The* unknown *section in every iteration includes all words not in the* known *section* $N_{tot} - N_\nu$.
In addition to the given corpus, an augmented corpus*, which is initially empty, is created. At the end of the* n *iterations the augmented corpus contains* n *modified copies of the original corpus.*

1: Compute the OOV rate ν of the cross validation set.
2: Estimate the number of words N_ν in the *known* corpus necessary to build a vocabulary with OOV rate ν measured on the *unknown* corpus (see Section 3.4.2).
3: $D = \gcd(N_\nu \pm \Delta N_\nu, N_{\text{tot}} - \Delta N_{\text{tot}})$.
4: $n = N_{\text{tot}}/D$
(In the example of Figure 5.1 $n = 3$, i.e., three iterations are shown.)
5: **LOOP** through Step 6 until the n iterations are complete
6: Make next division of original corpus into *known* and *unknown* sections.
7: Set up a vocabulary of all words in the *known* section of the corpus.
8: Replace all words in the *unknown* section which do not occur in the *known* vocabulary with the symbol `<UNK>`.
9: Append the *known* corpus and the modified *unknown* corpus to the augmented corpus. These two corpora together form a modified version of the original corpus.
10: **END LOOP**
The augmented corpus now effectively contains n *modified copies of the original corpus, where* n *is the number of iterations chosen.*
11: Append the original corpus to the augmented corpus.
12: Set up a list of all N-grams occurring in this final augmented corpus and how often each N-gram occurs.
13: Divide the N-gram counts by $n + 1$.

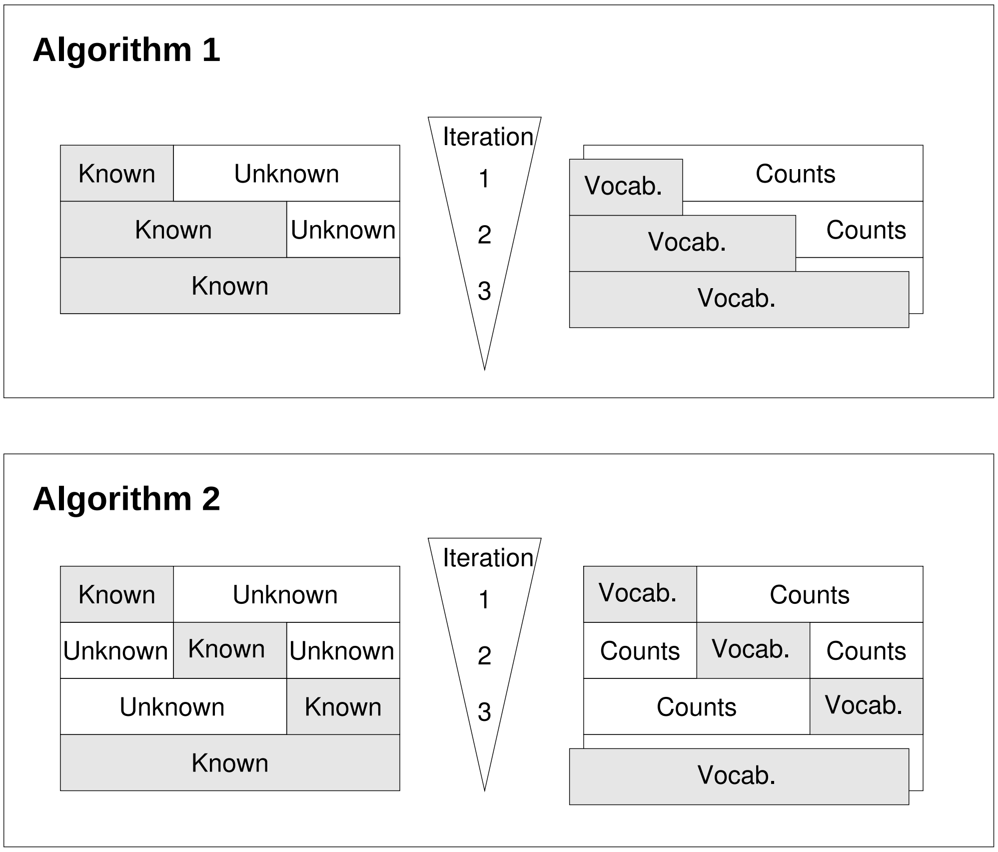


Figure 5.1: Graphical representation of the iterative substitution algorithm. The original algorithm introduced in [Fetter et al., 1995] is shown at the top, and the modified algorithm presented in this thesis is shown at the bottom. On the left, the division of the corpus into known (gray boxes) and unknown (white boxes) sections is shown. On the right, the gray boxes represent the section of the corpus used to build vocabularies, and the white boxes the section of the corpus used to compute N-gram counts. In this example, three iterations are shown ($n = 3$).

$$\log_2 \mathrm{PP}_{\to \mathtt{OOV}} \quad = \quad -\frac{1}{n} \sum_{\substack{i=1 \\ w_i \in \mathtt{OOV}}}^{n} \log_2 \mathrm{P}(w_i \mid w_{i-1}) \tag{5.7}$$

Equation 5.6 includes the probability contributions of transitions from OOV words to in-vocabulary words. Equation 5.7, on the other hand, measures the probabilities from transitions to OOV words.

We then ran the iterative substitution algorithm with different parameters, and obtained modified versions of the training texts with different OOV rates. Using these texts, several language models were trained and tested on the test set specified in Table 5.2.

Figure 5.2 shows the perplexities for different OOV rates in the training text. As can be seen, the more OOV words we have, the "easier" (i.e. more probable) it becomes for OOV words to be predicted by their predecessors (i.e. $\mathrm{PP}_{\to \mathtt{OOV}}$ decreases). On the other hand, it becomes "harder" for OOV words to predict their successors (i.e. $\mathrm{PP}_{\mathtt{OOV} \to}$ increases) simply because its probability distribution becomes "flatter." This tradeoff results in a U-shaped curve of the overall perplexity shown in the bottom plot of Figure 5.2. As expected, the minimum is located where the OOV rate of the training text is the same as the test text (1.62 %).

Iterative Substitution: Recognition Evaluation

To benchmark our progress, we first ran our recognizer without OOV-word models. The first row of Table 5.5 show the results. Then we selected a language model trained with texts processed by iterative substitution. In order to select one of the language models tested in Figure 5.2, we predicted the expected OOV rate of the test set using the method described in Section 3.4.2[9]. Since the overall language model performance remains relatively constant for training sets with OOV rates between 0.5 % and 2 %, as shown in Figure 5.2, the recognition and detection results do not vary significantly for any of the language models of this interval.

Finally, the recognizer was run using the OOV-word acoustic models presented in Section 5.3 and the selected iterative substitution language model. All other recognition parameters, including the language model scaling factor (ϑ in Equation 2.13) were the same as in the baseline system. Here, again, no measurable difference in the run time of these two systems was observed. Table 5.5 shows

[9] In this simulation, we also accounted for the fact that the test set consisted of whole dialogs, spoken exclusively by speakers that had not yet occurred in the training set. Therefore, the predicted OOV rate was higher than the numbers reported in Section 3.4.2. In addition, direct comparisons between Figure 3.4 and Figure 5.2 are misleading, since the former shows the OOV rate of the reference transliterations, whereas the latter illustrates the OOV rate as perceived by the language model.

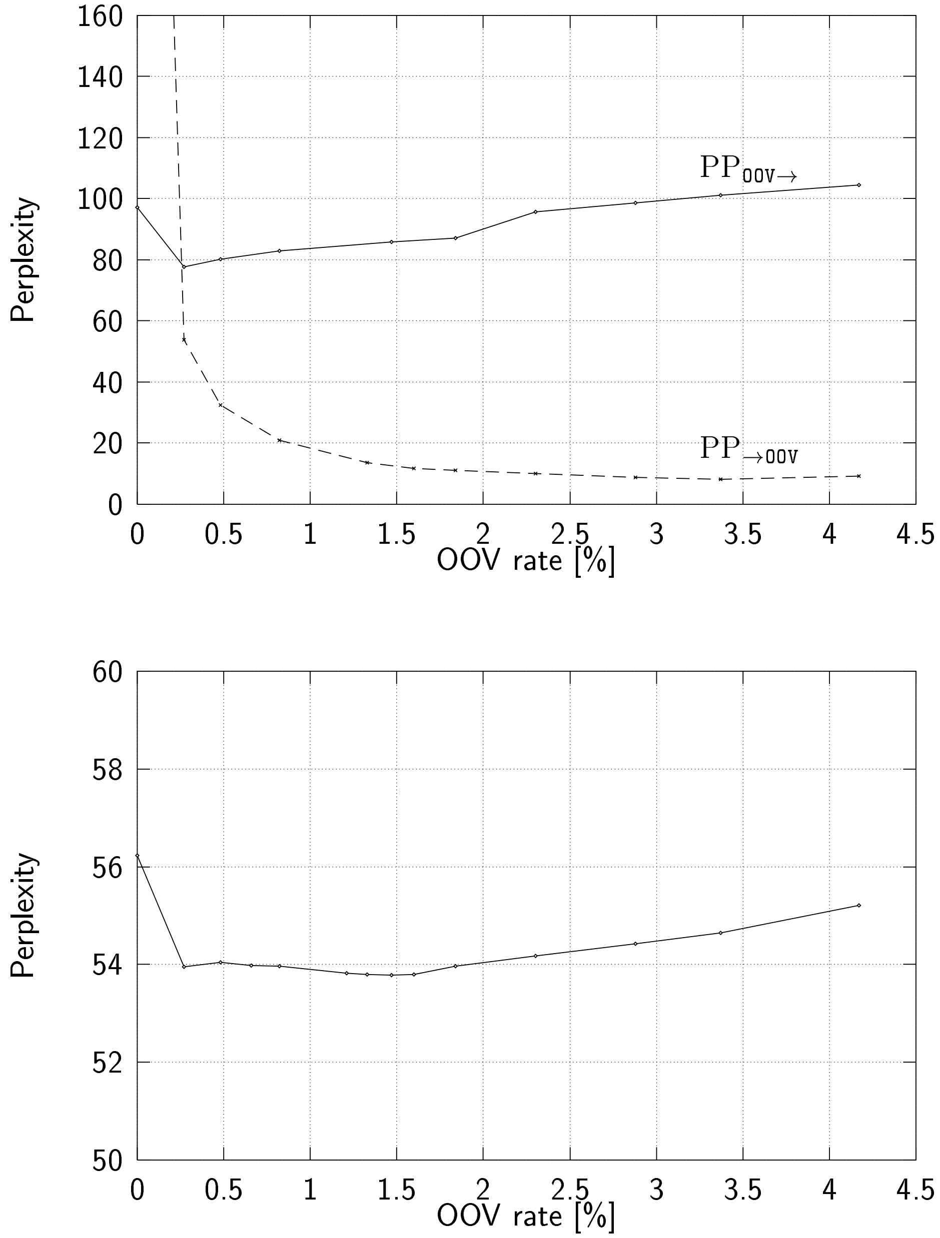


Figure 5.2: Perplexity in the vicinity of OOV words (top), and overall test set perplexity (bottom) plotted against the OOV rate of the language model training text. $PP_{OOV\rightarrow}$ is the perplexity of the bigrams that contain an OOV word in their history. $PP_{\rightarrow OOV}$ is the perplexity of the bigrams that directly involve OOV words.

that one fifth of all OOV words could be correctly detected. The precision rate indicates that every second OOV-word hypothesis was indeed correct. A small (although not significant) gain in word accuracy was achieved as well.

System	Recall	Precision	WA
No OOV modeling	0 %	—	77.6 %
OOV modeling and IS	21 %	49 %	77.7 %

Table 5.5: Baseline system without OOV-word modeling vs. a system with acoustic OOV-word models and iterative substitution (IS).

The recall rate is rather low, but considering the high precision rate, this result can be viewed as satisfactory, especially when taking the difficulty of the task into account. Furthermore, comparing Table 5.3, where real OOV words of a well defined class were available for language model training, with Table 5.5, where iterative substitution simulated OOV words, we see that the detection performance is indeed lower in the latter case, but still close to the former.

5.5 Error Analysis

A very interesting question in this context is related to the influence of the OOV rate on the performance of the system. In particular, we would like to know how many errors are caused by OOV words. In order to investigate this, two systems which differ only in their vocabularies and OOV rates were compared:

the "extended vocabulary" system, which was used as baseline in Section 5.4.2 and includes all words in the Verbmobil training corpus, and

the "reduced vocabulary" system, which was used as baseline in Section 5.4.1 and includes all words in the Verbmobil training corpus except for the proper names.

Table 5.6 summarizes the characteristics of both systems. As can be seen, an absolute increase of 2.15 % in the OOV rate produces an absolute performance deterioration of 4 %[10].

[10] Here we ignore the effects of the reduced confusability of the smaller vocabulary. Based on the conclusions arrived in [Rosenfeld, 1995c], and our own experimental obsevation, we consider the influence of this factor negligible.

System	Vocabulary	OOV rate	WA
Extended vocabulary	5,300	1.96 %	77.6 %
Reduced vocabulary	4,750	4.11 %	73.6 %
Δ	− 550	+ 2.15 %	− 4.0 %

Table 5.6: Performance comparison of the baseline of Section 5.4.2 with a larger vocabulary and lower OOV rate; and the baseline system of Section 5.4.1 which does not contain any proper names and consequently has a higher OOV rate.

Since none of the systems include any OOV-word modeling scheme, all OOV words are always misrecognized. This gives a lower bound of at least one error per OOV word:

$$\text{Errors per OOV word} \geq 1 \tag{5.8}$$

But how do we explain the 4 % decrease in accuracy, instead of 2.15 % (which would correspond to equality in Equation 5.8)? Here two factors related to the acoustic and the language models play an important role. On the one hand, the acoustic content of a OOV word can frequently be more accurately covered by a succession of short words than by just one word which is roughly as long as the OOV word in question. Therefore, OOV words tend to be substituted by sequences of in-vocabulary words, thus producing more than one error. Figure 5.3 illustrates this situation.

Reference	da	bin	ich	schon	in Münster
Extended vocabulary	da	bin	ich	schon	in Münster
Reduced vocabulary	da	bin	ich	schon	*mir ins hat*

Figure 5.3: Example of an OOV word being substituted by sequences of shorter words. The utterance *da bin ich schon in Münster* (I am already in Münster) is correctly recognized by the "extended vocabulary" system. The "reduced vocabulary" system did not contain the word *Münster* in its lexicon, and replaced it with three shorter words.

On the other hand, the language model also tends to propagate errors around OOV words. Whenever a word is misrecognized, the words around it will have the wrong context, and will therefore have a higher likelihood of being misrecognized. An example of this "domino" effect is illustrated in Figure 5.4.

Reference	aus	unserer	Filiale	in	Dresden	zurück
Extended vocabulary	aus	unserer	Filiale	in	Dresden	zurück
Reduced vocabulary	aus	unserer	Filiale	in	*den dreißigsten*	zurück

Figure 5.4: Example of the influence of the language model on the propagation of OOV-word errors. The partial utterance *aus unserer Filiale in Dresden zurück* (back from our branch in Dresden) is correctly recognized by the "extended vocabulary" system. The "reduced vocabulary" system did not contain the word *Dresden* in its lexicon, and replaced it by two words. Acoustically, the insertion of the function word *den* does not make much sense. But since the bigrams `[in den]` and `[den dreißigsten]` are extremely frequent in the training corpus, the language model preferred *in den dreißigsten* instead of, say, *in dreißigsten*.

In order to quantify the upper bound of the number of errors produced by OOV words, the references and the hypotheses were aligned as usual. Then, error sequences occurring in the surrounding of OOV words were counted, and the resulting number was divided by the total number of OOV words in the reference text. For both the extended and reduced vocabulary system, the ratio was approximately 2.5. Since this was confirmed in other experiments, we concluded that:

$$2.5 \geq \text{Errors per OOV word} \geq 1 \tag{5.9}$$

A histogram of the number of errors counted around OOV words was also computed. Figure 5.5 shows that most OOV words produce multiple errors. Only about one fifth of all OOV words produce only one recognition error. It can also be seen that two errors per OOV word is the most frequent response of our system.

Having bounded the number of errors produced by OOV words, we will now estimate the average of this number. For this, we applied the method used in the ARPA 1994 benchmark test [Pallett et al., 1995]. The test set is first divided into an "in-vocabulary" subset, which contains those turns consisting exclusively of in-vocabulary words; and an "out-of-vocabulary" subset, which also contains OOV words. The in-vocabulary subset is used to estimate the intrinsic error rate of the recognizer (due to factors other than OOV words). The out-of-vocabulary subset is assumed to have a total number of errors consisting of the sum of intrinsic errors and errors due to OOV words. The number of errors produced by OOV words is estimated subtracting the number of intrinsic error from the total number of errors in the out-of-vocabulary subset.

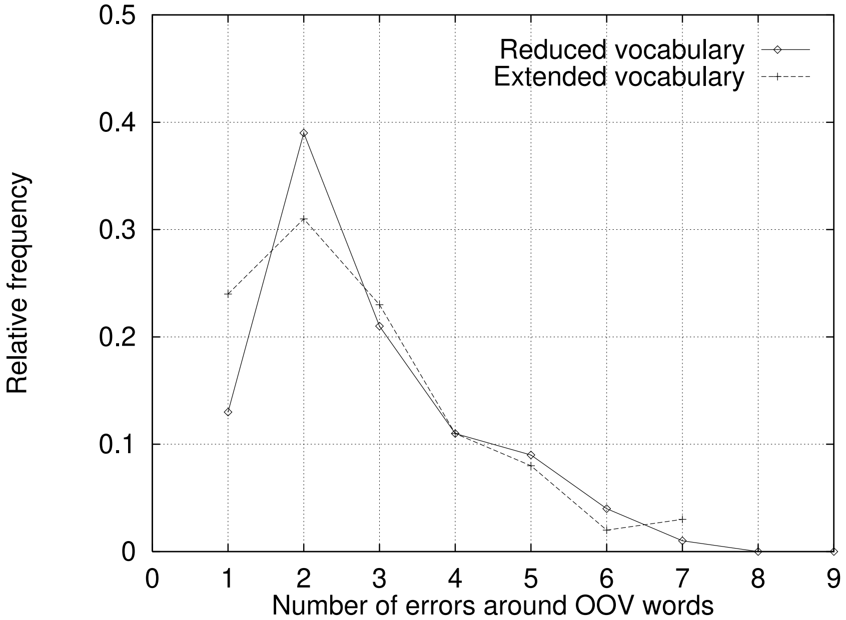


Figure 5.5: Histogram of the number of errors produced by OOV words.

Table 5.7 shows the results obtained. It can be seen that the performance degrades dramatically (approx. 10 % absolute) when the system is confronted with utterances that contain OOV words. The average number of errors per OOV word in our system lies between 1.8 and 1.9. This outcome is consistent with prior studies, that report averages ranging from 1.7 to 2.1 errors per OOV word for different systems [Pallett et al., 1995].

Both the extended and reduced vocabulary systems analyzed so far did not include any OOV-word modeling scheme. When the acoustic and language models for OOV words discussed in Sections 5.3 and 5.4 are used, the average number of errors is reduced from the 1.8 to 1.9 errors per OOV word (Table 5.7) to 1.5 to 1.7 errors per OOV word.

5.6 Summary

This chapter discussed methods for detecting OOV words. The detection approaches reviewed from the literature and our own work can be divided into *explicit* models (e.g. garbage modeling) and *implicit* modeling schemes (e.g. confidence measures). We opted for explicit models, and in particular word-based garbage models, because they provide a good trade-off between performance, simplicity,

System	IV / OOV	WA_{IV}	WA_{OOV}	Errors per OOV word
Extended vocabulary	63 % / 37 %	81 %	71 %	1.9
Reduced vocabulary	42 % / 58 %	81 %	69 %	1.8

Table 5.7: Summary of the results obtained estimating the number of errors per OOV word for both extended and reduced vocabulary systems. The test set is divided into two subsets: the In-Vocabulary (IV) and the Out-Of-Vocabulary (OOV) subsets. The proportion of the test set assigned to each subset is shown in the column labeled "IV / OOV." The word accuracy measured in each subset is shown in the columns WA_{IV} and WA_{OOV}.

and computational requirements.

The acoustic models for OOV-words were augmented with a language model especially tuned to detecting OOV words. Two different situations were analyzed. When OOV words occur in the language-model training material, the context of OOV words can be explicitly modeled. Recognition experiments showed that in this scenario system accuracy can be significantly improved by using models for OOV words. When all words in the training are included in the vocabulary, OOV words will not occur in the training material. In order to train a language model with OOV words we introduced a method dubbed *iterative substitution*, which simulates the occurrence of OOV words in the language-model training material. This method produced satisfactory experimental results.

Finally, the influence of OOV words on the system performance was investigated. It was found that most OOV words produce multiple errors, and that, on the average, the number of errors per OOV words is close to 2.

6 TRANSCRIBING OOV WORDS

OUR goal in this chapter is to find the phonetic transcription of spoken utterances, and in particular, OOV words. The transcriptions should be as accurate as possible, while at the same time minimize the amount of computation required. We present a method which uses information extracted directly from the word-based search to compute the most likely phoneme sequence. Utterances are transcribed during recognition, so that the phonetic representation of the input is available after the search. Using this method, the computational cost of the word-based search remains almost unaltered, and the phonetic transcription is obtained almost for free.

6.1 Introduction

Previously we examined different approaches to the problem of *detecting OOV words*. As mentioned before, OOV words affect a wide range of applications of speech recognition technology. In most cases the mere fact of detecting them (as opposed to confusing them with one or more in-vocabulary words), has a positive effect on the response of the system.

This chapter deals with another aspect of the OOV-word problem, namely the incorporation of detected OOV words into the vocabulary of the system. This subject is referred to as *learning new words*. In the following paragraphs, we will show that finding the phonetic description of OOV words is one of the most substantial issues when adding new words into the VERBMOBIL system. Throughout this thesis we follow the convention used in the VERBMOBIL project and reserve the term "transcription" for phonetic transcription only. Orthographic transcriptions are denominated "transliterations". However, we also use the redundant denomination "phonetic transcription" for clarity reasons.

When incorporating a new word into the speech recognizer, the following knowledge sources must be updated [Hetherington, 1995a, Hetherington, 1995b]:

- the acoustic models,

- the lexical (pronunciation) models, and
- the language model.

The modification of the first and last knowledge sources, namely the acoustic and language models, will not be treated in this thesis. The interested reader is referred to the excellent works by Hetherington [Hetherington, 1995a, Hetherington, 1995b] and Jelinek et al. [Jelinek et al., 1990] for studies on these subjects.

Whereas the detection problem can be treated in a generic fashion, without considering specific characteristics of the application, the learning problem is constrained by the application in many different ways. Below we will categorize application characteristics that are relevant to the learning problem.

A new word can be added to the system either as a whole-word model, or as a succession of sub-word units. In the former case it is indispensable to generate an acoustic model for the new word by retraining the whole-word model. Updating the lexical models is in this case trivial: The lexical representation of the new word is simply the designation of the whole-word model. In the latter case (i.e. when sub-word units are used to model the new word acoustically), which we are concerned with, the main problem consists of finding a lexical representation of the new word compatible with the existing pronunciation models of the system. This allows existing sub-word units to be used to describe the new word in the expanded lexicon.

A further important distinction between speech recognition applications, is whether the orthographic description of the new word can be obtained from the user or not. In dictation systems, for instance, the user can input the new word through the keyboard. Using the orthographic description of a new word greatly simplifies its incorporation in the system. If a keyboard input is not possible, as in the VERBMOBIL scenario, the situation is more complicated.

If the orthographic description of the new word is available, the phonetic transcription required to update the lexical models can be obtained by look-up in a large phonetic dictionary or by generating it using a text-to-speech system. If it is not available, the phonetic transcription must be extracted from the audio signal alone. This is the scenario in the VERBMOBIL project (see Section 3.3.2). There, two persons use the VERBMOBIL device as a communication aid. It is therefore absolutely necessary that the system be able to deal with OOV words, especially with the proper names of the persons involved in the dialog.

How does VERBMOBIL deal with new words? When the speech recognition module hypothesizes an OOV word, the system starts a clarification dialogue to confirm the hypothesis. If a proper name is confirmed, the new word should be incorporated into the vocabulary. For this, orthographic and phonetic transcriptions are needed. The (unknown) orthography of the word is represented by a dummy token (something like <NewWord01>), and the phonetic transcription is computed on-line, as explained below.

In this chapter we present a very efficient method for generating phonetic transcriptions in the VERBMOBIL scenario. We first discuss previous research work dealing with this subject. After that we will describe our transcription approach, and its implementation in the Daimler-Benz large-vocabulary continuous-speech recognition system. We will then present some experimental results and general conclusions.

6.2 Prior Work

6.2.1 Transcription of OOV Words

Asadi et al. presented the first approach towards detecting and transcribing OOV words [Asadi et al., 1991]. The detection approach was already discussed in Chapter 5. For transcription, they used both a phoneme recognizer, and a commercially available text-to-speech system. For the latter, they assumed that the orthography of the new word is available.

The authors worked with the DARPA 1000-Word Resource Management corpus of American English, which is distinguished by its particularly "clean" speech. Even then, they reported phoneme error rates that ranged between 44 % and 18 %, that is between 10 times and 3 times the word error rates of the system. Presumably, these numbers refer to already detected OOV words, which were repeated by the user and transcribed by a phonetic recognizer in a second run. Further, they found that the (phoneme) error rate of the text-to-speech system was significantly lower than the phoneme recognizer. The results obtained in re-recognizing the new words incorporated to the system vocabulary were very positive.

As mentioned in Chapter 5, Kita et al. presented a detection and transcription approach for OOV words based on two competing grammars: First, the *task grammar*, which encodes information on word sequences allowed, as well as its phonetic representation. And second, the *phonetic grammar*, which describes the phonological structure of Japanese[1] [Kita et al., 1991]. The system outputs a phonetic transcription of the spoken utterance for both in-vocabulary and out-of-vocabulary words.

Experiments were carried out with an extremely small corpus (which contained only 14 phrases). The authors reported very low phoneme error rates at the phrase level, where a phrase is counted as correct if all its phonemes were correctly

[1] This grammar was combined with a stochastic trigram model of syllables, which takes advantage of the relatively simple and regular structure of Japanese syllables.

transcribed. But since this measurement includes in-vocabulary words, which are determined by the task grammar, it is difficult to compare these numbers. Unfortunately, since the experimental conditions of this work are very constrained, we cannot draw general conclusions.

Itou et al. presented a similar approach in [Itou et al., 1992]. The allowed word sequences were determined by an LR parser for Japanese. A second grammar, which was in this case a phoneme trigram, was used to model OOV utterances, as discussed in Chapter 5.

The test database consisted of simple sentences generated by a grammar. For testing, some words were deleted from the vocabulary to simulate the occurrence of OOV words. Under these conditions, a very high detection rate of 75 % was obtained. The phoneme error rate of those OOV words correctly detected was 49 %. In order to reduce the amount of computation, the authors propose controlling the *beam width* of the phonetic search with a kind of confidence measure. Still, two search processes have to be run in parallel, which considerably increases the computational requirements of the recognizer.

The *new word model* presented in [Asadi et al., 1990] has also the ability of providing phonetic transcriptions. In [Asadi et al., 1991], however, it was used for OOV-word detection only, and other transcription methods were employed, as explained above. In [Young and Ward, 1993], a similar generic new word model was used for OOV-word detection *and simultaneously* OOV-word transcription. Unfortunately, though, no experimental results on this issue were reported.

Suhm et al. presented a detailed analysis of different attributes of OOV words in [Suhm et al., 1993] (see Chapter 3). In addition, they showed the importance of a language model, which is especially tuned for OOV word detection, as discussed in Chapter 5. For the acoustic models, they used an automaton which allowed an arbitrary sequence of phonemes, similar to the new word model proposed by Asadi et al. in [Asadi et al., 1990]. Further, a phoneme trigram model was implemented to reduce the search space and improve the phoneme transcription accuracy.

The approach was tested on a small read corpus, in which proper names were considered OOV words, and thus excluded from the vocabulary. The phoneme error rate measured on correctly detected OOV words was 62.6 %, or 5.5 times the word error rate. No specifications were given on the method used for transcription (a separate phoneme-based recognition run, or Asadi's new word model?). The computational costs of the technique were not reported either. Supposedly, though, the costs are high.

Hetherington presented in his Ph.D. thesis [Hetherington, 1995a] a detailed analysis of issues related to the incorporation of OOV words in different system components. An abridged version of the thesis was presented in [Hetherington, 1995b].

He investigated the importance of retraining the acoustic, lexical and language model in the ATIS task [Price, 1990]. For the acoustic model, he found that retraining does not necessarily improve the overall performance of the system. For the lexical models, he assumed that the phonetic description of the OOV words could be looked up in a dictionary, and concentrated on generating the pronunciation graphs needed by the SUMMIT system [Hetherington, 1995a]. Here again, retraining does not improve the overall performance of the system. Finally, his results show that language models do benefit from a retraining with samples of the new words.

In [Haeb-Umbach et al., 1995], Haeb-Umbach et al. deal with the problem of transcribing utterances from which many samples are available. They propose two different methods. In the first one, as many phonetic transcriptions are generated as samples of the word in question are available. Later, the transcription which has most likely generated all samples is selected. In the second method, an average transcription is computed using an HMM trained with all samples. This average transcription can also be used as an additional candidate for the first method. Also, the pronunciation given by a dictionary can be used in this manner.

Experiments were conducted with a German telephone corpus. The input words were first transcribed as explained above, and then incorporated into the lexicon. Finally, the system was tested using the same words, uttered by different speakers. Using very small lexica (10 and 32 words), "the error rates obtained with the acoustically-based automatic transcription methods are comparable to the error rates obtained with dictionary transcriptions" [Haeb-Umbach et al., 1995]. Unfortunately, no information on phoneme error rates was provided.

Our VERBMOBIL project partners at the Universities of Bielefeld and Karlsruhe in Germany concentrated on a detection approach based on phonotactic restrictions for German syllables [Jusek et al., 1995], [Kemp and Jusek, 1996], as discussed in Chapter 5. This approach has the positive side effect of providing the phonetic transcriptions of the detected OOV words. Again, the authors do not report phoneme error rates, so that a comparison with our work in terms of accuracy is impossible. Since in this approach OOV-word detection and transcription are tackled simultaneously, the computational price seems to be quite expensive. The phonotactic network is extremely complicated, and contains over 5,000 states and 15,000 transitions. Even when this approach is implemented in a simple way, as in [Kemp and Jusek, 1996], all states in the phonotactic network have to be included in the lexicon, thus increasing the search space considerably.

Matsunaga et el. used a two-pass strategy for OOV word detection and transcription [Matsunaga and Sakamoto, 1996]. In the first pass, coarse models were used to distinguish between in-vocabulary and OOV words. In the second pass, more specialized models were used for phonetic transcription. This approach was chosen in order to reduce the computational requirements of running two different model sets.

Tests were carried out on a very small Japanese database. As in [Kita et al., 1991], the reported phoneme error rate (75.2 %) includes in-vocabulary words, which are constrained by the task grammar. Therefore, we can make no judgment on the quality of this technique. Nevertheless, this work shows similarities to ours with respect to a controlled evaluation of the computational cost required by the different variants of the method.

6.2.2 Dictionary Learning

Lately, an increasing amount of effort has been dedicated to the topic of *dictionary learning*. Dictionary learning describes the aim of automatically obtaining word pronunciations from the training data. This makes the time-consuming and error-prone process of manually transcribing data unnecessary. It also allows statistically significant pronunciation variants, which are extremely important in spontaneous speech, to be systematically and consistently determined.

For dictionary learning the output of a phoneme recognizer is usually matched against a forced alignment of the training material. This generates a first set of pronunciation models. Afterwards, these pronunciations can be further processed and training of the acoustic models can be repeated. For the subject of study in this chapter the experience in phoneme recognition is interesting.

Sloboda introduced this topic in [Sloboda, 1995]. He presented a very simple method and successfully tested it within the VERBMOBIL task. Further refinements of the technique were presented in [Sloboda and Waibel, 1996]. Unfortunately, the accuracy of the phonetic recognizer used was not evaluated.

Fosler et al. [Fosler, 1996] made a DP alignment of the base pronunciations to a hand-labeled subset of the Switchboard corpus. They measured a phoneme error rate of 35.2 % (and a phoneme deletion rate of 12.5 %).

Dictionary learning and new word learning are very related subjects. Notably, when the orthographic description of the OOV word is available, the differences between the two methods vanish. If the orthographic descriptions are not available, the methods are discerned by their difficulty. In dictionary learning, a precise segmentation of the spoken words can be easily obtained (through forced alignment of the given reference texts to the spoken input). In new word learning, the segmentation has to be performed on hypotheses which might contain errors. This makes the problem of finding accurate phonetic transcriptions much more difficult in the new word learning scenario we are concerned with.

6.3 The Cheap Transcription Algorithm

Most of the previous research on OOV word transcription reported in the literature have the following characteristics in common:

- Two separate search processes are used (one for the word-based search and one for the phoneme-based search), and consequently,
- the computational cost of transcribing OOV words is high.

In the current study, we present a phonetic-transcription method which uses information extracted directly from the word-based search to compute the most likely phoneme sequence. Consequently, a second search is avoided, and little additional computation is required for the phoneme transcription. Therefore, this method, was dubbed *the cheap transcription algorithm*.

Since we view *detection* and *transcription* of OOV words as two separate problems, we opted for two separate strategies. The cheap transcription algorithm does not assume knowledge of detected OOV sections in the input signal. On the contrary, the whole utterance is transcribed during the (word-based) search, so that the phonetic representation of the input is always available after the search. Because this transcription approach is independent of OOV-word detection, it can be presented and evaluated without taking the detection aspect into account. Hence, in the next section we present our approach in a phonetic-typewriter fashion, i.e. as an algorithm intended to solely find the phonetic representation of the spoken input at a low cost. But we do not loose sight of the main goal, namely using these transcriptions to phonetically describe OOV words. In Section 6.4 we concentrate on this aspect of the problem.

Before going into the details of the cheap transcription algorithm, we will say a few words on the structure of the recognizer lexicon, which will provide the reader with necessary background information. As mentioned in Chapter 2, the lexicon is implemented as a tree in the Daimler-Benz speech recognition system. This tree is compiled off-line and can be directly accessed and traversed during decoding. In the tree, every node is a (pointer to an) HMM state. For isolated word recognition, one could perform the search directly on the lexicon tree. For continuous speech recognition, the situation is more complicated, since a new lexicon can be started anytime. Also, all active paths are expanded in every frame so that the static lexicon structure becomes a highly dynamic, growing tree (see Figure 6.1 for an example). Here, the search for the best path is performed by means of the Viterbi algorithm. (See Section 2.2.3 for further details.)

Keeping this picture in mind, it is intuitively clear that the phonetic information encoded in the Hidden Markov Models is accessible during the lexical search.

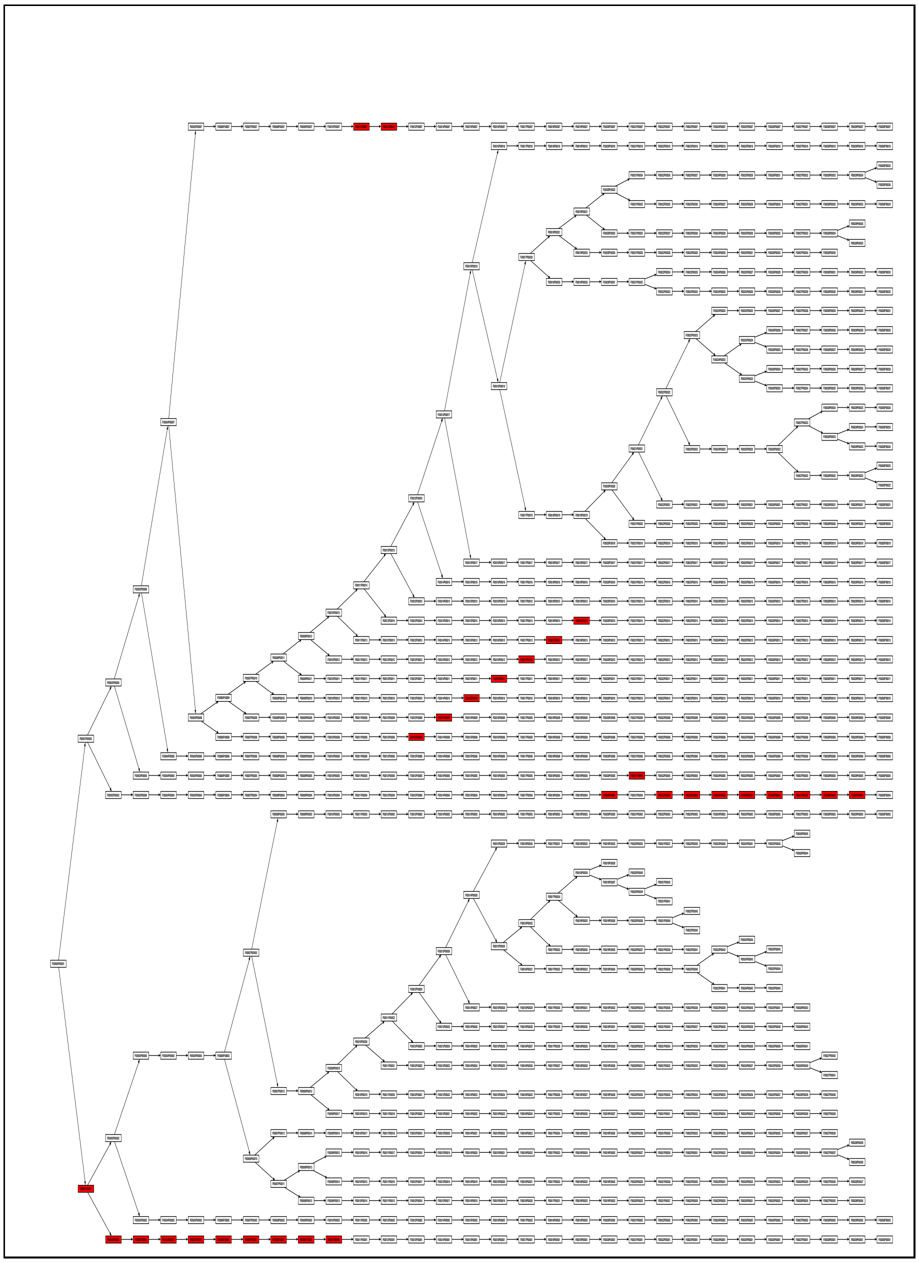


Figure 6.1: The search space of the first 30 frames (0.3 sec) in a VERBMOBIL sentence. There, the word *Wiedersehen* (good bye) was uttered. Every column in the graph represents a frame and every node an HMM state. The nodes marked in black correspond to the best path in the current frame.

As mentioned before, every node in the lexicon tree represents one state of an HMM. And every HMM is again one context-dependent subword unit. So for phonetic transcription we store the best lexicon node in every frame (for more on the meaning of "best lexicon node" see Section 6.3.1 below). Afterwards, we map the context-dependent subword unit into the corresponding context-independent model, which is one in a set of 36 German phonemes[2] in the system used for this study.

At the end of the decoding process, we obtain a chain consisting of as many phonemes, as (10 msec) frames in the utterance. In order to obtain the actual phonetic transcription, this chain is first smoothed and then collapsed into a set of phonemes. For smoothing we employed a simple heuristic based on a sliding weighted window, as will be explained in more detail in Section 6.3.1. Figure 6.2 shows an example of this method.

Phoneme chain:

```
U  o: U  o: o: o: o: o: k k k k k k k k k k e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: f
```

Smoothed phoneme chain:

```
o: o: o: o: o: o: o: o: k k k k k k k k k k e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e:
```

Smoothed and collapsed phoneme chain:

```
o:---------------------k------------------e:------------------------------------------------------
```

Figure 6.2: Smoothing and collapsing frame-based chains. In this example, the word *okay* (transcribed `/?o:ke:/` in German) was uttered. The first row shows the phonemes extracted in each frame during the search. The second row shows the smoothed phoneme chain. In the third row, the smoothed chain was collapsed into the string `/o:ke:/`. In this case, the glottal stop (`?`) was not recognized.

We must emphasize that the algorithm ignores some of the constraints and knowledge sources that make word-based recognition possible at all. The most obvious knowledge source ignored is the lexicon. But also the constraints imposed to the search by the topology of the HMMs are ignored: Whereas a word hypothesis has to fully traverse all HMMs it includes in order to survive, the phonemes hypothesized by the cheap transcription algorithm do not.

Moreover, the effectiveness of the algorithm is restricted by the influence of two characteristics of the system: the vocabulary size and the beam width. The

[2]The phoneme set used for this study includes the following SAMPA symbols [Fourcin et al., 1989]:

```
2: 6 9 ? @ C E I N O S U Y a b d e: f g h i: j k l m n o: p r s t u: v x y: z
```

In addition, silence and noise were modeled in the system, but not considered for evaluation.

influence of the vocabulary size is obvious. Think of the extreme case when the lexicon contains only one word: The transcription would be constrained to the phonemes present in this one word, regardless of what was actually said. But in large vocabulary systems, the high phonemic diversity present in the lexicon lessens the importance of this drawback.

The influence of the beam width is less evident. As mentioned in Chapter 2, the beam threshold is a continuously updated minimal value for pruning less probable paths. The interval between the path with the highest probability and the pruning threshold is called the *beam width.* If the beam width is reduced to zero, that is, if the only path allowed is the best one, the transcription would be constrained to the phonetic representation of that best path. But since in large-vocabulary, continuous-speech recognition systems the beam width *has* to be fairly wide to achieve good recognition results, this problem is, again, of less practical importance.

In spite of these drawbacks, the benefits of the method are so appealing that it is worthwhile investigating it further. Using the cheap transcription algorithm the computational cost of the word-based search remains almost unaltered, so a phonetic transcription is obtained almost for free (more on this in Section 6.3.5). As a bonus, this method does not require any changes in the system lexicon, since the transcription is extracted directly from the models competing in the search.

6.3.1 Preliminary Experiments

For the following experiments, the recognizer used as the baseline in Section 5.4.2 was employed. The acoustic and language models were trained with the material defined for the Verbmobil Evaluation 1996, as summarized in Table 6.1 (see also [Reinecke, 1996]). The vocabulary contains approximately 5,300 words and includes all words in the training material (see page 105 for more details).

Experiments were carried out using the Kiel Corpus of Spontaneous Speech (Vol. I) [Kohler et al., 1995], [Kohler, 1996]. The corpus is a subset of the Verbmobil corpus, and contains 21 dialogs in 383 turns (approx. 40 min of speech). The material was manually segmented and labeled at the phoneme level, and can therefore be used as a reference to match the recognizer's frame-wise phoneme transcriptions[3]. For our tests we used a subset of the corpus, comprising approximately 17 min of speech, as shown in Table 6.1. (The remaining portion of the corpus was reserved for training a phoneme trigram model, as explained below.)

Corpus	Duration	Dialogs	Turns	Words
Training	28 hours	633	12,001	$\approx$ 268 K
Test	17 min	21	199	$\approx$ 2.9 K

Table 6.1: General statistics of the training and test subsets of the VERBMOBIL corpus used for the recognition experiments presented in this chapter.

In order to first optimize the approach without taking the influence of OOV words into account, the test set was included in the training set. Thus the OOV rate of the test set was artificially reduced to 0 % for the purpose of this study. Also, the bigram language model used during the search had a perplexity of only 48, because all bigrams in the test set were observed in the training. Consequently, an artificially high word recognition performance was obtained, namely 81.9 %. In Section 6.3.6 we will further study the response of the system under more realistic testing conditions.

In separate recognition runs, additional information intended to be used for phonetic transcription was extracted from the search. Two different parameters were investigated:

- the phonemes belonging to the best path in every frame (BPF), and

- the phonemes with the highest (product of HMM transition and emission) probability in every frame (BMF).

In both cases, no measurable difference in the run time was observed with respect to the original system, as will be discussed below.

[3]We compared the canonical phoneme references of the test set with the phoneme references actually spoken, as provided by the Kiel Corpus of Spontaneous Speech. This was achieved with the standard DP procedure described in Chapter 2, which aligns the references minimizing the Levenshtein distance [Lehning, 1994a]. We found a mismatch in 18 % of all phonemes. Deletions made up the major proportion of those mismatches (78 % of all mismatches, or 14 % deletion rate). This clearly shows the statistical importance of elisions in spontaneous speech, as well as the difficulty of bridging the gap between the phoneme and word levels. Since our intention is just to evaluate the results of the phoneme recognizer, a careful analysis of this subject is beyond the scope of our work. In the rest of this chapter, we will use the actually spoken phoneme references provided in the Kiel Corpus of Spontaneous Speech.

6.3.2 Frame-Based Experiments

The two phoneme chains investigated were compared to the reference chains on a frame basis. Every frame with a hypothesis different from the spoken phoneme was counted as an error[4]. The *Frame Error Rate* (FER) was calculated as follows:

$$\mathrm{FER} = \frac{\mathrm{Count}(\mathtt{ERR})}{\mathrm{Count}(\mathtt{REF})} \tag{6.1}$$

where Count(...) represents the number of frames containing incorrect phoneme hypotheses (ERR) and the total number of frames in the test set (REF).

Table 6.2 shows the results. As can be seen, the phonemes that belong to the best path per frame (BPF) provide the best transcription. This outcome is somehow intuitive: The best path per frame has a degree of inertia, that compensates the lack of constraints mentioned earlier. On the other hand, the best instantaneous phoneme (BMF) is not subject to any stabilizing force, and can therefore vary widely from frame to frame. Since the inferiority of the latter parameter was confirmed in further experiments, we abandon it at this point.

System	FER
Best path per frame (BPF)	38.82 %
Best model per frame (BMF)	44.99 %

Table 6.2: Frame error rates (FER) for the best path per frame (BPF), and best model per frame (BMF).

In addition, we extracted from the search not only the phoneme chain belonging to the *best* path in every frame (BPF), but also all distinct symbols belonging to paths within a given beam width. Consequently, the number of alternative phonemes outputted varies from frame to frame. Using a frame-based phoneme language model, the most likely phoneme sequence can be extracted similarly to the way the best word sentence is extracted from a word graph, as explained in Section 2.2.4. This sequence is different to the one belonging to the best path in every frame (BPF), because the context is also considered for the decision on the identity of the current phoneme.

The back-off trigram language model [Clarkson and Rosenfeld, 1997] was trained with the half of the Kiel Corpus of Spontaneous Speech not used for

[4] A few frames were not considered for evaluation, because the references were labeled with symbols for diphthongs which we do not consider as a separate acoustic model. Also, silence and non-speech segments in the references were skipped.

testing. It contains 184 turns and approximately 150 K symbols (that is, 150 K frames, or 25 min). The perplexity of the model measured on the test set was 1.96. As can be seen in Figure 6.3, increasing the number of hypotheses per reference phoneme considered to extract the most likely phoneme sequence improves the quality of the frame-based phoneme chain. This clearly shows the influence of the context provided by the language model. Since this approach appears to be promising, we will continue investigating it in our phoneme-based experiments.

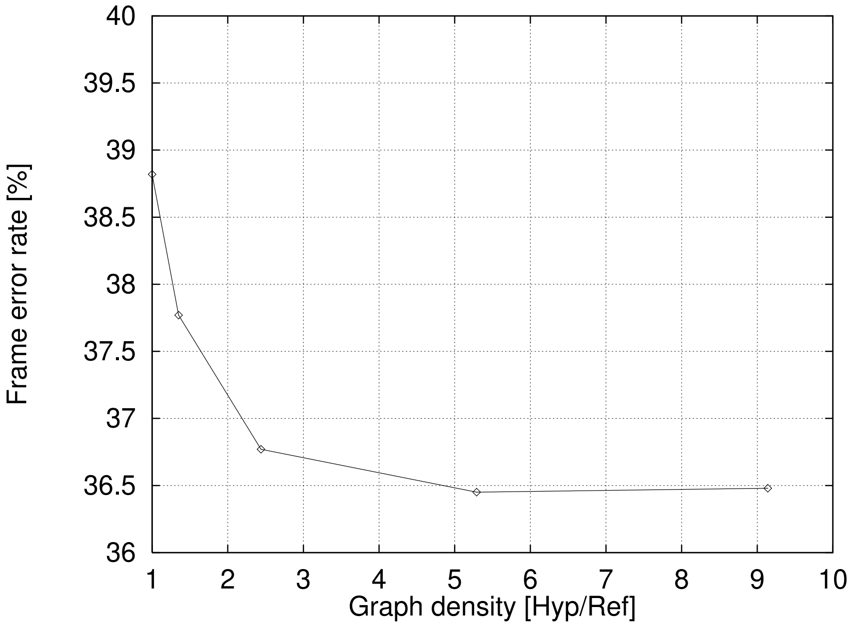


Figure 6.3: Frame error rates of the most likely phoneme sequence. This phoneme sequence is extracted from a graph structure containing a number of alternatives per spoken frame (Graph density).

6.3.3 Phoneme-Based Experiments

In order to obtain the actual phonetic transcription, the frame-based chains are smoothed and then collapsed. Collapsing consists of simply reducing successions of identical tokens to one token. For smoothing, a sliding weighted window was used: To decide on the identity of a phoneme, all symbols within an interval given by the window width are considered. The weighted frequency of every symbol inside the window is computed. The weight is determined by an exponential function of the distance of the symbol in question to the window center (symbols further apart contribute less). The symbol with the highest weighted frequency replaces the phoneme in the center of the window.

After the phoneme chains were smoothed and collapsed, the *Phoneme Error Rate* (PER) was measured:

$$\mathrm{PER} = \frac{\mathrm{Count}(\texttt{INS}) + \mathrm{Count}(\texttt{SUB}) + \mathrm{Count}(\texttt{DEL})}{\mathrm{Count}(\texttt{REF})} \tag{6.2}$$

where Count(...) represents the number of insertions (INS), substitutions (SUB), and deletions (DEL) among the hypothesized phonemes, when aligned to the reference phonemes (REF) in the test set[5].

Results are presented in Figure 6.4. The plots show the optimization of the two free parameters in the smoothing algorithm: the exponential weighting function on the abscissa, and the window width parametric to the curves. As can be seen, smoothing has a very positive effect, reducing the error rate considerably. Moreover, the smoothing parameters show a relatively stable behavior. A phoneme error rate of slightly below 34 % was obtained.

Next, we experimented with BPF chains containing a variable number of alternative phonemes per frame n, with $n \geq 1$. As mentioned above, the most likely phoneme chain ($n = 1$) was extracted using a frame-based phoneme language model. This chain was then smoothed and collapsed, and finally aligned to the spoken phoneme references. Figure 6.5 shows the results. A very significant improvement of the phoneme accuracy was obtained: The phoneme error rate was reduced from $\approx$34 % to $\approx$30 %. It is interesting to note that as the number of alternatives presented to the language model grows, the importance of smoothing decreases: In this case, the language model does the smoothing.

6.3.4 Baseline System

For comparison purposes, the Daimler-Benz speech recognizer was also run in phonetic recognition mode. In this mode, phonemes are treated as if they were words. These "words" can be either context-independent (CI) or context-dependent (CD) HMM recognition units. In the former case, the lexicon consists of the 36 phonetic symbols used throughout this chapter, plus two models for silence and noise. In the latter case, we used approximately 1500 different phonemes in context. In addition, we also performed experiments with phoneme bigram back-off language models [Clarkson and Rosenfeld, 1997], which were used for constraining

[5] In order to calculate *frame* error rates, no alignment is necessary because the strings to be compared have the same length. For *phoneme* error rates, the alignment with the smallest Levenshtein distance is used, as explained in Chapter 2.

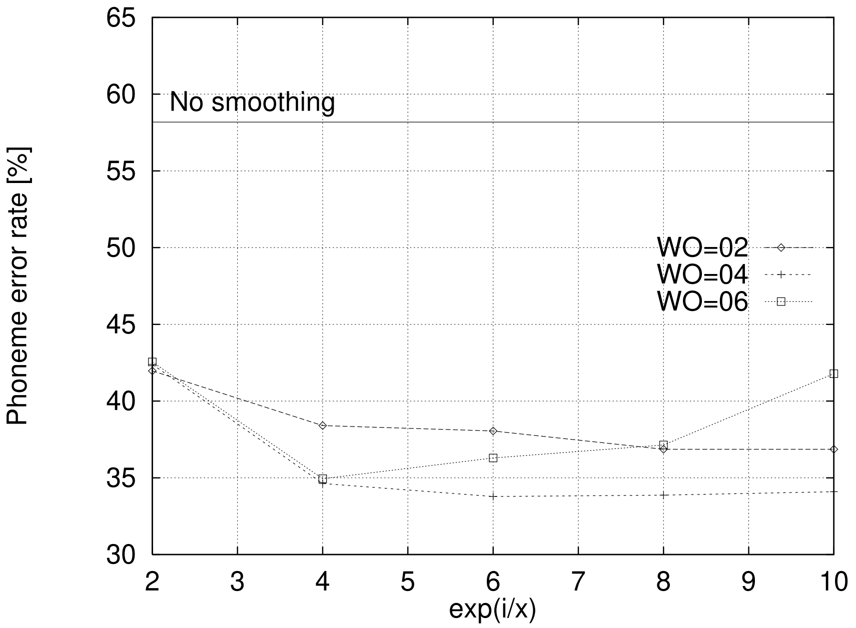


Figure 6.4: Phoneme error rate for different smoothing parameters: The exponential weighting function $\exp(i/x)$ is shown on the abscissa. i is the index of the elements within the window being smoothed, and x is the smoothing parameter to be empirically optimized. The window offset WO is shown parametric to the curves. The window width WW can be calculated as follows: $\mathrm{WW} = 1 + 2\mathrm{WO}$.

the search. The context-independent bigram language model was trained with the half of the Kiel Corpus of Spontaneous Speech not used for testing[6]. For the context-dependent bigram language model the whole VERBMOBIL corpus (excluding development and test sets) was used. Table 6.3 summarizes the characteristics of the language models. It is interesting to note that the language model of context dependent units (CD) is much more robust, even though its vocabulary is much larger. This is due to the strong contextual constrains imposed by context dependent units.

The results obtained for different system configurations are shown in Table 6.4. As expected, the performance of the recognizer with context-dependent units (CD) is significantly better than with context-independent units (CI). Further, Table 6.4 clearly shows the importance of using a phoneme bigram language model. The

[6]Tests with other training sets were performed as well. The bigram trained with the actually spoken phoneme strings from the Kiel Corpus of Spontaneous Speech yielded the lowest perplexity and was therefore chosen for recognition experiments.

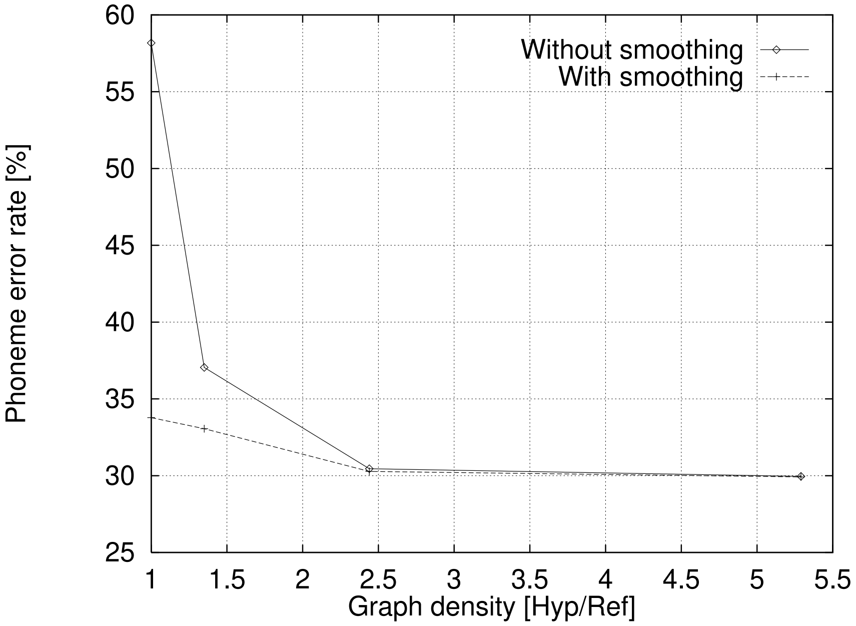


Figure 6.5: Phoneme error rates of the most likely phoneme sequence. This phoneme sequence is extracted from a graph structure containing a number of alternatives per spoken frame (Graph density).

language model is especially advantageous for large vocabularies, as in the context-dependent case. The best performance obtained is ≈28 % phoneme error rate using context-dependent units and a bigram language model. This outcome is comparable to the ones reported by other VERBMOBIL partners [Heine et al., 1997].

6.3.5 Comparison of Performance

Next, a comparative analysis of the computational resources needed by the different transcription methods presented above was performed. This analysis is

LM	No. of turns	No. of tokens	Vocab. size	PP
CI	184	20 K	38	13
CD	9,600	1 M	1,500	4

Table 6.3: Characteristics of the language models used for the baseline phoneme recognizers.

System	`INS`	`SUB`	`DEL`	PER
CI models, no LM	9 %	29 %	8 %	48.03 %
CI models, bigram	10 %	25 %	6 %	42.17 %
CD models, no LM	6 %	25 %	9 %	41.95 %
CD models, bigram	5 %	12 %	9 %	27.79 %

Table 6.4: Baseline phoneme error rate (PER) for different system configurations: First, context-independent (CI) models are tested without and then with a phoneme bigram language model. Afterwards context-dependent models (CD) are also tested in a similar manner. In addition, insertion (`INS`), substitution (`SUB`), and deletion rates (`DEL`) are reported.

not intended as a comparison of the Daimler-Benz speech recognizer with other systems reported in the literature. For this, carefully controlled experiment conditions with identical hardware and software platforms are required. Our goal is simply to see the reported phoneme error rate in terms of the required computation time.

For the different methods tested, we measured the *elapsed time* needed to complete a recognition run. Then, we normalized them by the duration of the test set to obtain the *real time factor* (RTF). We report here the overhead RTF, which represents the additional time required to generate a phonetic transcription. All experiments were conducted on a Digital AlphaServer 8200 5/300 (SPECfp95 11.7, as reported in `http://open.specbench.org/osg/cpu95/results/cfp95.html`). Our word recognizer runs in 1.3 times real time, that is RTF ≈ 1.3.

Table 6.5 summarizes our measurements. The cheap transcription algorithm allows the input utterance to be transcribed with no (BPF) or very small computational overhead (BMF with frame trigram). In both cases, the phonetic transcriptions are more accurate and less expensive than those obtained using context-independent models with a phoneme recognizer. The lowest error rates are still obtained with a phoneme recognizer using context-dependent units and a phoneme bigram language model. Yet this performance gain is very costly (note that overhead RTF ≈ 6) due to the high confusability of the vocabulary (context-dependent phonemes), which makes pruning techniques ineffective. For comparison, we also tuned the recognizer parameters to allow more pruning. We could accelerate the context-dependent baseline recognizer from overhead RTF ≈ 6 to overhead RTF ≈ 3 with a minor performance degradation (PER $= 30.97\,\%$). Still, the transcriptions provided by the cheap transcription algorithm are a better compromise between performance and computational costs.

System	overhead RTF	PER
Baseline: CI models, no LM	≈ 0.4	48.03 %
Baseline: CI models, bigram	≈ 0.5	42.17 %
Baseline: CD models, bigram	≈ 6	27.79 %
Cheap transcription: BPF	≈ 0	33.78 %
Cheap transcription: BPF, frame trigram	≈ 0.2	29.92 %

Table 6.5: Overhead real time factor (RTF) and phoneme error rate (PER) for different transcription methods. The baseline systems are phoneme recognizers with different configurations. The cheap transcription algorithm is evaluated with and without a language model.

6.3.6 Influence of the System Accuracy

So far the word recognizer with the cheap transcription algorithm was a closed system, that is, the test material was included in the training material. This was done intentionally so as to be able to analyze the performance of the method without taking the effect of OOV words into account. In order to test this system under more realistic conditions, and also to quantify the influence of the word error rate on the phoneme error rate provided by the cheap transcription algorithm, the experiments were repeated using an open system. The open system is exactly the same as the closed system used so far, except that the test set was excluded during the training of the models. The vocabulary was constructed, as before, using all words in the training material, and consequently did not include those words occurring only in the test set.

The results obtained testing both systems are summarized in Table 6.6. As expected, the test set perplexity increases dramatically, because many bigrams in the test set were not observed in the training. (The bigram hit rate was 86 %, a typical number for the VERBMOBIL corpus.) Consequently, also the word accuracy in the open system is much lower (over 5 % absolute degradation) than that in the closed system.

Since, as explained above, the cheap transcription algorithm is embedded in the word search, search errors (i.e. recognition errors) are likely to have a negative influence on the phoneme recognition rate. In order to determine the magnitude of this influence, we repeated all phoneme-based experiments presented in Section 6.3.3 with the BPF chains extracted from the open system.

After extracting the most likely phoneme chain, smoothing, and collapsing, we obtained a surprisingly positive result: The increase of over 5 % in word error rate increased the phoneme error rate less than 1 %, as can be seen in Figure 6.6.

System	Vocabulary	OOV rate	PP	WA
Closed	5327	0.00 %	48	81.9 %
Open	5261	1.12 %	86	76.8 %

Table 6.6: Comparison of performance of a closed system, which included the test material in the training, and an open system, which did not. Vocabulary size, OOV rate, bigram perplexity and word accuracy of both systems are shown.

Thus the cheap transcription algorithm appears to be robust against changes in the word accuracy of the system it is embedded in.

6.4 Transcribing Out-Of-Vocabulary Words

In the previous section we introduced the cheap transcription algorithm and tested its ability to phonetically transcribe speech. Phonetic transcriptions of speech are necessary in a number of speech recognition applications. They can be used among other things to compute confidence measures, to synthesize OOV words, or to incorporate new words into the system vocabulary. In this section we will concentrate on the phonetic transcriptions of OOV words, which can be used to incorporate them into the lexicon.

If a spoken utterance contains OOV words, and if these OOV words can be detected by some means, their phonetic description can be computed using the transcription of the whole utterance, and the segmentation (beginning and end times) of the detected OOV word. In this case, new words could be incorporated into the system fully automatically without involving the speaker in the process. This, however, would require very high detection and precision rates, as well as a precise segmentation of the spoken words. In today's technology this is only feasible for very constrained tasks, for example when using a deterministic grammar for recognition.

Unfortunately, in the Verbmobil task, high OOV-word detection and precision rates cannot be guaranteed. Moreover, if an OOV word is detected, its exact beginning and ending times (needed to extract the corresponding phonemes from the transcription of the whole utterance) are difficult to determine accurately. Therefore, we decided to test the performance of our OOV-word transcription approach independently of the detection results presented in Chapter 5. In the following we will therefore assume correct detection, and concentrate on the transcription accuracy alone.

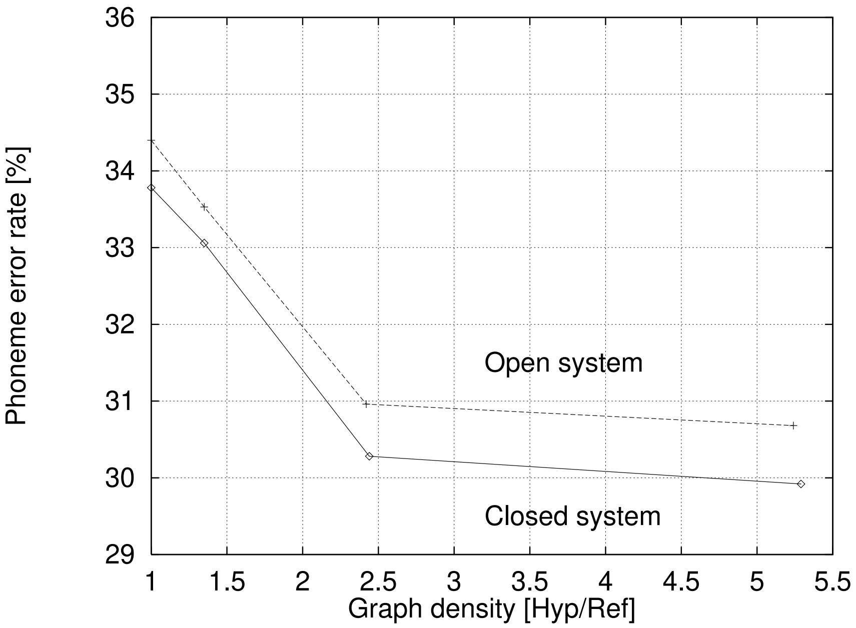


Figure 6.6: Phoneme error rates of the most likely phoneme sequence for both the closed system and the open system. This phoneme sequence is extracted from a graph structure containing a number of alternatives per spoken frame (Graph density).

To measure the transcription accuracy of OOV words, the phoneme error rate was measured not on the whole utterance as before, but only between the boundaries of OOV words, which are those words not included in the vocabulary of the open system introduced in Section 6.3.6. The exact boundaries of the OOV words used for reference were taken from the Kiel Corpus of Spontaneous Speech.

Figure 6.7 shows the results. It can be seen that the difference in phoneme error rate between the baseline system and the cheap transcription system with closed vocabulary is very similar to that reported in Section 6.3.5. The baseline system with context dependent models performs only 1.5 % better than the cheap transcription system. In these two systems, however, OOV words do not have any influence, either because the spoken words are irrelevant (the vocabulary of the baseline system consists of phonemes), or because there are no such OOV words (the cheap transcription system is a closed vocabulary system).

In the open cheap transcription system, on the other hand, the phoneme error rate is measured on out-of-vocabulary sections of the utterance. Since the word search will produce errors in these sections, we expected the performance of the cheap transcription algorithm to decrease significantly. Figure 6.7 shows a dete-

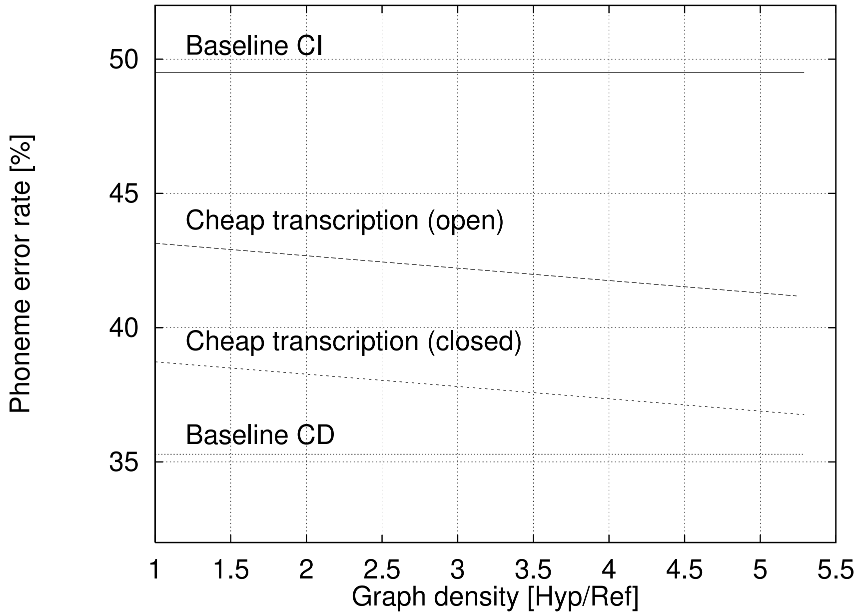


Figure 6.7: Phoneme error rates obtained using the cheap transcription algorithm measured against the baseline system. The cheap transcription was first obtained for a closed system (where the phoneme error rate is measured on words included in the vocabulary) and then for an open system (where the phoneme error rate is measured on OOV words). In both cases, the most likely phoneme sequence is extracted from a structure containing a number of alternatives per spoken frame (Graph density). Results of the baseline system are given for context independent (CI) and context dependent (CD) models.

rioration of approximately 4.5 % with respect to the closed cheap transcription system, or approximately 6 % with respect to the baseline system. This difference may seem very large at first glance, but considering that roughly 60 % of the correct phonetic transcription could be extracted from wrong recognition hypotheses, these results are very positive.

6.5 Summary

We presented a method dubbed *the cheap transcription algorithm*, which can be used to transcribe utterances during recognition, so that the phonetic representation of the input is available as a by-product of the (word-based) search. Using this method, the computational cost of the word-based search remains almost unaltered. As a bonus, the method does not require any changes in the system lexicon, since the transcription is extracted directly from the models competing in the search. Furthermore, the phonetic transcriptions obtained using the cheap transcription algorithm are almost as accurate as those obtained with a phoneme recognizer at a much higher cost.

When transcribing OOV words, the cheap transcription algorithm suffers from the errors made during the word search. The transcription performance, however, remains reasonably good even under these difficult conditions.

7 SUMMARY AND CONCLUSIONS

THIS thesis analyzes the significance of *Out-Of-Vocabulary words* in speech recognition and suggests several ways of alleviating the problem.

7.1 Summary

The significance of any problem (and thus OOV words) depends on two factors: first, how often it occurs and second, how much the overall system goals can be improved if the problem can be solved. To answer the first question, the relationship between corpus size and vocabulary size were studied for several German corpora (Chapter 3, page 39). As expected, these two parameters are highly correlated and strongly depend on the characteristics of the corpus being considered.

A study of the coverage of different vocabulary sizes showed that the OOV rate of unseen test material can be significantly reduced by increasing the size of the vocabularies. Yet even with very large vocabularies, the OOV rate remains high in all corpora studied. In the VERBMOBIL corpus, for example, the OOV rate obtained when using the whole corpus for building the vocabulary (except for the test set) ranges between 1 % and 2 % at the word level. On the turn level, however, it was about 20%, i.e. about one in every five turns contained on the average an OOV word. This is clearly much too frequent to ignore.

The second question, namely how the system can be improved, is very much related to the question of how OOV words can be handled. Basically out-of-vocabulary words can be eliminated by including them in the vocabulary, thus making them *in-vocabulary* words, or they may be tagged as a joker, which is essentially saying that the recognizer realizes that it does not recognize this acoustic sequence as anything known to it. This information can then be passed on to the higher (linguistic) levels of the recognition system.

The most trivial way to eliminate OOV words is to increase the recognition vocabulary. This was attempted using information extracted from other corpora and application domains (Chapter 3, page 50). Experiments showed that the coverage of "cross-domain" vocabularies was, as expected, much lower than the coverage of "intra-domain" vocabularies. When complementing domain-specific vocabularies with cross-domain vocabularies, however, the coverage could be significantly improved. This is especially valid when the original domain-specific vocabularies are small.

A second method of reducing the OOV rate has to do with the definition of "recognition units" or words. The ultimate objective of a speech recognition system for any language is to identify the acoustical units of a spoken utterance. Once this has been achieved, the results can be used in a number of ways. One such way is to express the results "in writing[1]." This presupposes some standardized orthography for the language. In English, the orthographic concept of word correspond pretty much one to one with these "acoustical units." In German, however, the orthographic phenomenon of writing two or more semantically independent units together is quite prevalent, thus leading to a very large (and potentially infinite) number of compound words (Chapter 4, page 68).

This deficiency can be remedied by redefining the concept of *word* to correspond to semantic units. If the final measure of correctness is a correct *orthographic* representation of the utterance, however, then a mechanism must be found to recombine the semantic units to orthographically correct words. We proposed using automatically derived morphologically-oriented lexical units instead of (classical) words. In particular, compound words are first decomposed into their constituent parts, which are used for recognition, and then the parts are rejoined to compounds. These units reduce the vocabulary size as well as the OOV rate, and improve the recognition performance at the same time (Chapter 4, page 72).

A corpus study showed further that OOV compound words are mostly made up of in-vocabulary components. These OOV words could be "recovered" by reassembling them from their component parts. In order to achieve this, various statistics on the composition of compound words were collected from the training set and used to rejoin in-vocabulary components to out-of-vocabulary compounds (Chapter 4, page 80). Unfortunately, this approach did not yield satisfactory results. Our conclusion is that the investigated corpus statistic metrics are not sufficient to model compound-word building.

Since it will never be possible to construct a lexicon with all words, eliminating OOV words in effect only postpones the ultimate question of what to do with those remaining OOV words which cannot be included in the lexicon in advance. When OOV words do occur, they are erroneously assigned to other words in the lexicon

[1] It is also possible to identify these acoustical units with numbers 1 through n, for instance, and use these results in a dialog system, with no recourse whatsoever to any orthographic representation.

(if no prevision for OOV words is made in the system). To make things worse, OOV words tend to produce multiple errors due to the contextual constraints applied during recognition. On the average, our recognizer makes approximately two errors per OOV word (Chapter 5, page 112).

There are essentially two methods for identifying OOV words in an acoustic utterance: implicit and explicit. Implicit methods utilize confidence measures of the hypotheses derived from the lexicon containing known words. Explicit methods attempt to model OOV words as one or more acoustical models which then reside in the lexicon as full partners with the other in-vocabulary words. The grammar then treats these OOV word models simply as an additional class of words which may occur in certain contexts.

Even though implicit models can be very beneficial for many speech recognition applications (e.g. [Fetter et al., 1996a]), they intrinsically lack specificity as "OOV-word detectors," because the primary aim of these methods is not to detect OOV words as such, but rather to detect portions of the input signal that do not match existing models well, and thus produce recognition errors. Since in VERBMOBIL OOV words are a minor source of errors, confidence measures are not an effective strategy for detecting OOV words (Chapter 5, page 98). Investigations for this thesis have shown that explicit models, and in particular word-based garbage models, provide a good trade-off between performance, simplicity, and computational requirements (Chapter 5, page 100).

This acoustic modeling scheme for OOV-words was augmented with a language model especially tuned to detecting OOV words (Chapter 5, page 102). This can be easily implemented when OOV words occur in the language-model training material. In this case, OOV words can be explicitly modeled with the OOV words occurring in their natural contexts. Using these models, approximately one third of all OOV words were correctly detected, and the system accuracy was significantly improved.

The detection rate obtained seems to be rather low at first glance. Two aspects of the detection problem should be discussed to explain this. First, the detection rate can be calibrated by changing the number of OOV hypothesized. In general, the more OOV words are hypothesized, the higher the detection rate becomes. But at the same time, a higher number of OOV hypotheses has a negative effect on the precision rate (see Equation 2.24), and hence on the overall word accuracy. We preferred to hypothesize OOV words sparingly since we are not willing to compromise the overall system performance to obtain higher detection rates. This way, precision rates which are even higher than the word accuracy were obtained (which means that OOV word hypotheses are more likely to be correct hypotheses than in-vocabulary words), and at the same time, the word accuracy was significantly increased when using OOV-word models.

The second aspect worth mentioning is the fact that the acoustic novelty present in OOV words is very low: Only 10 % to 20 % of the acoustic sequences contained in OOV words are new (Chapter 3, page 48). Indeed, most OOV words which were not correctly detected were very similar to one or more in-vocabulary words. Considering this, the results can be viewed as satisfactory, especially when taking the difficulty of the task into account.

When all words in the corpus are included in the vocabulary, OOV words will obviously not occur in the training material. This is the case in many real-world applications of speech recognition. Under these conditions, a language model will typically assign low probabilities to OOV words, which are based on coarse estimates. In order solve this problem we introduced a method dubbed *iterative substitution*, which simulates the occurrence of OOV words in the language-model training material. Using this method approximately one fifth of all OOV words were correctly detected, and the word accuracy improved slightly.

Comparing the previous scenario, where real OOV words of a well defined class were available for language model training, with the current one, where iterative substitution simulated OOV words, we see that the detection performance is indeed lower in the latter case, but still very close to the former. Therefore, this result can be viewed as positive.

The next step after identifying OOV words as such is to identify their phonetic content. If this is possible (even on a restricted basis), then it becomes possible to identify new words which occur more than once, such as proper names. This information can be of use in the higher recognition levels.

The traditional method of obtaining a phonetic transcription consists in using a separate search process. In this case, the computational cost of transcribing OOV words is high. In order to reduce the computational cost of the phonetic transcription, we presented a method dubbed *the cheap transcription algorithm*, which uses information extracted directly from the word-based search to compute the phonetic transcription. In effect, the phonetic representation of the input is available as a by-product of the (word-based) search, and consequently, little additional computation is required for the phonetic transcription.

Experiments showed that the phonetic transcriptions obtained using the cheap transcription algorithm are almost as accurate as those obtained with a phoneme recognizer at a much higher cost. When transcribing OOV words, the cheap transcription algorithm suffers from the errors made during the word search. The transcription performance, however, remains reasonably good even under these difficult conditions: Roughly 60 % of the correct phonetic transcription could be extracted from wrong recognition hypotheses.

7.2 Contributions of This Thesis

In general terms, the main contributions of this thesis are a detailed study of the problem of OOV words, as well as a number of solutions to this problem, ranging from vocabulary optimization to acoustic and language modeling, and phonetic transcription of OOV words. In particular, the following algorithms and methods are novel, and were proposed in this thesis:

- the use of information extracted from other application domains to optimize vocabularies,
- the use of morphological information to recover OOV words,
- the Iterative Substitution Algorithm, and
- the Cheap Transcription Algorithm.

7.3 Outlook

This thesis answered some questions related with the OOV-word problem, but at the same time it also raised some new ones. Also, some important and extensive topics could not be investigated at all due to time restriction. In the following paragraphs we will discuss (and also speculate) on those issues which we believe should be investigated in future work.

As discussed above, OOV words could be avoided, and consequently the importance of the OOV-word problem could be significantly reduced by building vocabularies that cover user utterances well enough. In Chapter 3, words extracted from other corpora and application domains were used to supplement the vocabulary in question. Those "cross-domain" words were selected using a frequency criterion.

Other methods should be investigated in future work. For instance, *cross-domain perplexity* measurements, as introduced in [Fetter et al., 1996b], can be used to preselect cross-domain sentences before applying the frequency criterion. Preliminary experiments using this method showed consistent improvements in the vocabulary coverage. Other information theoretical measurements like mutual information could be used for this purpose as well (see, for instance, [Jelinek, 1990]).

In Chapter 4 morphological knowledge was incorporated into the recognition process. Automatically derived morphologically-oriented lexical units were used to reduce the vocabulary size and improve the recognition performance. In an attempt to rejoin in-vocabulary components to OOV compounds, various statistics on the composition of compound words were used. Unfortunately, this approach did not yield satisfactory results. Our conclusion is that the investigated corpus statistic metrics are not sufficient to build a precise generative model of compound-words.

One possible extension of this work could be adding formal morphological knowledge. One of the main problems of knowledge-based approaches is their integration into a stochastic framework. The recent trend towards stochastic methods in computational linguistics may provide a solution for this in the next few years. Besides, the incorporation of higher knowledge sources such as semantic or domain knowledge should also be investigated in future work.

For detecting OOV words, several possible extensions of this thesis are worth mentioning. One obvious extension is the improvement of the detection parameters. This could be achieved by a combination of explicit and implicit detection methods. Some kind of dynamic weighting of each detection parameter should also be developed to obtain optimal detection performance.

The experience collected in this thesis, though, has clearly shown that the problem of OOV words cannot be solved by acoustic means alone (see, for instance, the discussion on acoustic novelty above). Even a "classical" stochastic language model does not provide a complete solution. We are convinced that a breakthrough can only be achieved by incorporating higher knowledge sources. They should support the detection process by providing syntactic, semantic, pragmatic, and domain information (see, for instance, [Young, 1994]). These knowledge sources can be integrated either directly into the recognition process or as a postprocessing module. Again, the particular implementation of this may come from the field of computational linguistics.

The next step after the system detects OOV words as such is to learn them. If this is possible, new words which occur more than once (e.g. proper names) could be recognized by the system as any other in-vocabulary word. It is desirable that new words be incorporated into the system fully automatically without involving the speaker in the process (i.e. unsupervised learning). This, however, would require very high detection performances. And even if this problem can be solved (see above), many other questions remain unanswered. The subject of new word learning has yet to be thoroughly investigated.

For learning a new word, the recognition system has to update its entire set of models. In this thesis only one aspect of the problem was studied: updating the pronunciation dictionary. For this, a phonetic transcription of the new words has to be found, and *the cheap transcription algorithm* was presented. In this study, being the first step towards learning new words, many simplification were met. For instance, when measuring the accuracy of the phonetic transcription, all errors were considered equal. If the phonetic transcription is used to incorporate a new word into the system lexicon, this is clearly wrong. The effect of errorful transcriptions (and other factors also) on the re-recognition performance has to be carefully studied in future work.

Finally, in order to achieve a good performance in new word learning, higher knowledge sources will inevitably have to be incorporated as well. This, of course, adds more complexity, since more models have to be updated for adding new words into the system.

Bibliography

[Asadi and Leung, 1993] Asadi, A. and Leung, H. (1993). New-word addition and adaptation in a stochastic explicit-segment speech recognition system. In *Proc. ICASSP'93*, pages 624–625, Toronto, Ontario, Canada.

[Asadi et al., 1990] Asadi, A., Schwartz, R., and Makhoul, J. (1990). Automatic detection of new words in a large vocabulary continuous speech recognition system. In *Proc. ICASSP'90*, pages 125–128, Albuquerque, New Mexico, USA.

[Asadi et al., 1991] Asadi, A., Schwartz, R., and Makhoul, J. (1991). Automatic modeling for adding new words to a large-vocabulary continuous speech recognition system. In *Proc. ICASSP'91*, pages 305–308, Minneapolis, Minnesota, USA.

[Baggia et al., 1994] Baggia, P., Gerbino, E., Gianchin, E., and Rulent, C. (1994). Experiences in spontaneous speech interaction with a dialog system. In Niemann, H., Mori, R. D., and Hanrieder, G., editors, *Progress and Prospects of Speech Research and Technology*, Proceedings in Artificial Intelligence, pages 241–248. Infix, Munich, Germany.

[Baumgärtner et al., 1973] Baumgärtner, K., Fritz, G., Herrlitz, W., Hundsnurscher, F., Kastowski, D., König, E., Külwein, W., Kürschner, W., von der Mülbe, U., Narr, B., Wagner, K.-H., and Weber, H. (1973). *Funk-Kolleg Sprache*, volume 1. Fischer Taschenbuch, Frankfurt, Germany. In German.

[Berton, 1996] Berton, A. (1996). Wortsegmentierung mittels morphologischer Informationen in der Sprachverarbeitung. Master's thesis, University of Chemnitz, Chemnitz, Germany. In German.

[Berton et al., 1996a] Berton, A., Fetter, P., and Regel-Brietzmann, P. (1996a). Behandlung von Komposita in Spracherkennungssystemen mit großem Vokabular. In *ITG-Fachtagung Sprachkommunikation*, Frankfurt, Germany. In German.

[Berton et al., 1996b] Berton, A., Fetter, P., and Regel-Brietzmann, P. (1996b). Compound words in large-vocabulary German speech recognition systems. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Bhatt, 1991] Bhatt, C. (1991). *Einführung in die Morphologie.* Gabel Verlag. In German.

[Bleiching, 1995] Bleiching, D. (1995). Morphologie in der Lexikondatenbank. Verbmobil technisches Dokument 20, Universität Bielefeld, Bielefeld, Germany. In German.

[Boite et al., 1993] Boite, J.-M., Bourlard, H., D'Hoore, B., and Haesen, M. (1993). A new approach towards keyword spotting. In *Proc. EUROSPEECH'93*, pages 1273–1276, Berlin, Germany.

[Boros et al., 1997] Boros, M., Aretoulaki, M., Gallwitz, F., Nöth, E., and Niemann, H. (1997). Semantic processing of out-of-vocabulary words in a spoken dialogue system. In *Proc. EUROSPEECH'97*, Rhodos, Greece.

[Brietzmann et al., 1994] Brietzmann, A., Class, F., Ehrlich, U., Heisterkamp, P., Kaltenmeier, A., Mecklenburg, K., Regel-Brietzmann, P., Hanrieder, G., and Hiltl, W. (1994). Robust speech understanding. In *Proc. ICSLP'94*, pages 967–970, Yokohama, Japan.

[Bub and Schwinn, 1996] Bub, T. and Schwinn, J. (1996). Verbmobil: The evolution of a complex large speech-to-speech translation system. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Bub et al., 1997] Bub, T., Wahlster, W., and Waibel, A. (1997). Verbmobil: The combination of deep and shallow processing for spontaneous speech translation. In *Proc. ICASSP'97*, volume 1, page 71, Munich, Germany.

[Carter et al., 1996] Carter, D., Kaja, J., Neumeyer, L., Rayner, M., Weng, F., and Wiren, M. (1996). Handling compound nouns in a Swedish speech-understanding system. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Clarkson and Rosenfeld, 1997] Clarkson, P. R. and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge toolkit. In *Proc. EUROSPEECH'97*, Rhodos, Greece.

[Class et al., 1991] Class, F., Kaltenmeier, A., and Regel, P. (1991). Soft-decision vector quantization based on the Dempster/Shafer theory. In *Proc. ICASSP'91*, pages 665–668, Toronto, Ontario, Canada.

[Class et al., 1993a] Class, F., Kaltenmeier, A., and Regel-Brietzmann, P. (1993a). Evaluation of an HMM speech recognizer with various continuous speech databases. In *Proc. EUROSPEECH'93*, pages 1587–1590, Berlin, Germany.

[Class et al., 1993b] Class, F., Kaltenmeier, A., and Regel-Brietzmann, P. (1993b). Optimization of an HMM-based continuous speech recognizer. In *Proc. EUROSPEECH'93*, pages 803–806, Berlin, Germany.

[Class et al., 1992] Class, F., Kaltenmeier, A., Regel-Brietzmann, P., and Trottler, K. (1992). Fast speaker adaptation combined with soft vector quanization in an HMM speech recognition system. In *Proc. ICASSP'92*, pages 461–464, San Francisco, California, USA.

[Cox and Rose, 1996] Cox, S. and Rose, R. C. (1996). Confidence measures for the Switchboard database. In *Proc. ICASSP'96*, volume 1, page 511, Atlanta, Georgia, USA.

[Dugast et al., 1995] Dugast, C., Aubert, X., and Kneser, R. (1995). The Philips large-vocabulary recognition system for American English, French and German. In *Proc. EUROSPEECH'95*, volume 1, pages 197–200, Madrid, Spain.

[Eckert et al., 1995] Eckert, W., Nöth, E., Niemann, H., and Schukat-Talamazzini, E. (1995). Real users behave weird—Experiences made collecting large human-machine-dialog corpora. In Dalsgaard, P., Larsen, L. B., Boves, L., and Thomsen, I., editors, *Proc. of the ESCA Tutorial and Research Workshop on Spoken Dialogue Systems*, pages 193–196, Vigsø, Denmark. ESCA.

[Eide et al., 1995] Eide, E., Gish, H., Jeanrenaud, P., and Mielke, A. (1995). Understanding and improving speech recognition performance through the use of diagnostic tools. In *Proc. ICASSP'95*, volume 1, page 221, Detroit, Michigan, USA.

[Fach, 1996] Fach, M. (1996). Optimierung von Wortuntereinheiten bei der Erkennung gesprochener Sprache. Master's thesis, University of Stuttgart, Stuttgart, Germany. In German.

[Fetter et al., 1995] Fetter, P., Class, F., Haiber, U., Kaltenmeier, A., Kilian, U., and Regel-Brietzmann, P. (1995). Detection of unknown words in spontaneous speech. In *Proc. EUROSPEECH'95*, pages 1637–1640, Madrid, Spain.

[Fetter et al., 1996a] Fetter, P., Dandurand, F., and Regel-Brietzmann, P. (1996a). Word graph rescoring using confidence measures. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Fetter et al., 1996b] Fetter, P., Kaltenmeier, A., Kuhn, T., and Regel-Brietzmann, P. (1996b). Improved modeling of OOV words in spontaneous speech. In *Proc. ICASSP'96*, pages 534–537, Atlanta, Georgia, USA.

[Fosler, 1996] Fosler, E. (1996). Automatic learning of word pronunciation from data. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Fourcin et al., 1989] Fourcin, A. J., Harland, G., Barry, W., and Hazan, V., editors (1989). *Speech Input and Output Assessment*. Ellis Horwood, Chichester.

[Fukunaga, 1972] Fukunaga, K. (1972). *Introduction to Statistical Pattern Recognition*. Academic Press.

[Furui, 1986] Furui, S. (1986). Speaker–independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59.

[Gallwitz et al., 1996] Gallwitz, F., Noeth, E., and Niemann, H. (1996). A category based approach for recognition of out-of-vocabulary words. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Geutner, 1995] Geutner, P. (1995). Using morphology towards better large–vocabulary speech recognition systems. In *Proc. ICASSP'95*, pages 445–448, Detroit, Michigan, USA.

[Godfrey et al., 1992] Godfrey, J., Holliman, E., and McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. In *Proc. ICASSP'92*, volume 1, pages 517–520.

[Good, 1953] Good, I. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264.

[Haeb-Umbach et al., 1995] Haeb-Umbach, R., Beyerlein, P., and Thelen, E. (1995). Automatic transcription of unknown words in a speech recognition system. In *Proc. ICASSP'95*, volume 1, page 840, Detroit, Michigan, USA.

[Hayamizu et al., 1993] Hayamizu, S., Itou, K., and Tanaka, K. (1993). Detection of unknown words in large vocabulary speech recognition. In *Proc. EUROSPEECH'93*, pages 2113–2116, Berlin, Germany.

[Heine et al., 1997] Heine, H., Jost, U., and Evermann, G. (1997). What is wrong with the lexicon—An attempt to model pronunciations probabilistically. In *Proc. EUROSPEECH'97*, Rhodos, Greece.

[Hetherington, 1995a] Hetherington, I. L. (1995a). *A Characterization of the Problem of New, Out-Of-Vocabulary Words in Continuous-Speech Recognition and Understanding*. PhD thesis, MIT, Cambridge, Massachusetts, USA.

[Hetherington, 1995b] Hetherington, I. L. (1995b). New words: Effect on recognition performance and incorporation issues. In *Proc. EUROSPEECH'95*, pages 1645–1648, Madrid, Spain.

[Hetherington and Zue, 1993] Hetherington, I. L. and Zue, V. (1993). New words: Implications for continuous speech recognition. In *Proc. EUROSPEECH'93*, pages 2121–2124, Berlin, Germany.

[Hirschberg and Nakatani, 1993] Hirschberg, J. and Nakatani, C. (1993). A speech-first model for repair identification in spoken language systems. In *Proc. EUROSPEECH'93*, pages 1173–1176, Berlin, Germany.

[Huang et al., 1990] Huang, X., Ariki, Y., and Jack, M. (1990). *Hidden Markov Models for Speech Recognition*. Information Technology Series. Edinburgh University Press, Edinburgh, Scotland, UK.

[Hübener et al., 1996] Hübener, K., Jost, U., and Heine, H. (1996). Speech recognition for spontaneously spoken German dialogues. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Hwang, 1997] Hwang, K. (1997). Vocabulary optimization based on perplexity. In *Proc. ICASSP'97*, volume 2, page 1419, Munich, Germany.

[Itou et al., 1992] Itou, K., Hayamizu, S., and Tanaka, H. (1992). Detection of unknown words and automatic estimation of their transcriptions in continuous speech recognition. In *Proc. ICSLP'92*, Banff, Alberta, Canada.

[Jelinek, 1990] Jelinek, F. (1990). Self-organized language modeling for speech recognition. In Waibel, A. and Lee, K.-F., editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, San Mateo, CA, USA.

[Jelinek et al., 1990] Jelinek, F., Mercer, R. L., and Roukos, S. (1990). Classifying words for improved statistical language models. In *Proc. ICASSP'90*, pages 621–624, Albuquerque, New Mexico, USA.

[Jusek et al., 1995] Jusek, A., Fink, G., Rautenstrauch, H., and Sagerer, G. (1995). Detection of unknown words and its evaluation. In *Proc. EUROSPEECH'95*, pages 2107–2110, Madrid, Spain.

[Kaltenmeier, 1991] Kaltenmeier, A. (1991). *Modellbasierte Worterkennung in Spracherkennungssystemen für großen Wortschatz*. PhD thesis, Technische Universität Berlin, Berlin, Germany. In German.

[Katz, 1987] Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35:400–401.

[Kemp and Jusek, 1996] Kemp, T. and Jusek, A. (1996). Modelling unknown words in spontaneous speech. In *Proc. ICASSP'96*, pages 530–533, Atlanta, Georgia, USA.

[Kita et al., 1991] Kita, K., Ehara, T., and Marimoto, T. (1991). Processing unknown words in continuous speech recognition. *IEICE Trans.*, E74(7):1811–1816.

[Kohler et al., 1994] Kohler, K., Lex, G., Pätzold, M., Scheffers, M., Simpson, A., and Thon, W. (1994). Handbuch zur Datenaufnahme und Transliteration in TP14 von Verbmobil, V3.0. Verbmobil technisches Dokument 11, Institut für Phonetik und digitale Sprachverarbeitung, Universität Kiel, Kiel, Germany. In German.

[Kohler et al., 1995] Kohler, K., Pätzhold, M., and Simpson, A. (1995). From scenario to segment: The controlled elicitation, transcription, segmentation and labelling of spontaneous speech. Technical Report 29, Institut für Phonetik und digitale Sprachverarbeitung, Universität Kiel, Kiel, Germany.

[Kohler, 1996] Kohler, K. J. (1996). Labelled data bank of spoken standard German: The Kiel Corpus of Read/Spontaneous Speech. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Kowtko and Price, 1989] Kowtko, J. and Price, P. (1989). Data collection and analysis in the air travel planning. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 119–125, Cape Cod, Massachusetts, USA.

[Kuhn, 1995] Kuhn, T. (1995). *Die Erkennungsphase in einem Dialogsystem*. PhD thesis, Universität Erlangen-Nürnberg, Sankt Augustin, Germany. In German.

[Kuhn et al., 1996] Kuhn, T., Fetter, P., Kaltenmeier, A., and Regel-Brietzmann, P. (1996). DP-based wordgraph pruning. In *Proc. ICASSP'96*, Atlanta, Georgia, USA.

[Lamel et al., 1995] Lamel, L., Adda-Decker, M., and Gauvain, J. L. (1995). Issues in large vocabulary, multilingual speech recognition. In *Proc. EUROSPEECH'95*, volume 1, pages 185–186, Madrid, Spain.

[Lee, 1989] Lee, K.-F. (1989). *Automatic Speech Recognition: The Development of the SPHINX System*. Kluwer Academic Publishers, Boston, Massachusetts, USA.

[Lehning, 1994a] Lehning, M. (1994a). Ein Programmsystem zur Evaluierung der signalnahen Spracherkennung. Verbmobil technisches Dokument 9, Technische Universität Braunschweig, Braunschweig, Germany. In German.

[Lehning, 1994b] Lehning, M. (1994b). Evaluierungsergebnisse der signalnahen Spracherkennung (Sommer 1994). Verbmobil Memo 25, Technische Universität Braunschweig, Braunschweig, Germany. In German.

[Lehning, 1995] Lehning, M. (1995). Spezifikationen für die Evaluierung der signalnahen Spracherkennung im Sommer 1995. In German. Available at `http://sbvsrv.ifn.ing.tu-bs.de/evaluierung`.

[Linde et al., 1980] Linde, Y., Buzo, A., and Gray, R. M. (1980). An algorithm for vector quantizer design. *IEEE Trans. on Communications*, 28(1):84–95.

[Lüngen et al., 1996] Lüngen, H., Pampel, M., Drexel, G., Gibbon, D., Althof, F., and Schillo, C. (1996). Morphology and speech technology. In *Proceedings of the ACL-SIGPHON Workshop on Computational Phonology and Speech Technology*, Santa Cruz, California, USA.

[Mangold et al., 1978] Mangold, H., Stall, D., and Zelinski, R. (1978). Sprach-Ein- und Ausgabe. Technical report, BMFT-Programm zur Förderung der Datenverarbeitung, Signal- und Mustererkennung, Ulm, Germany. In German.

[Matsunaga and Sakamoto, 1996] Matsunaga, S. and Sakamoto, H. (1996). Two-pass strategy for continuous speech recognition with detection and transcription of uknown words. In *Proc. ICASSP'96*, volume 1, page 538, Atlanta, Georgia, USA.

[McNair and Waibel, 1994] McNair, A. and Waibel, A. (1994). Improving recognizer acceptance through robust, natural speech repair. In *Proc. ICSLP'94*, pages 1299–1302, Yokohama, Japan.

[Meliani and O'Shaughnessy, 1996] Meliani, R. E. and O'Shaughnessy, D. (1996). New efficient fillers for unlimited word recognition and keyword spotting. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Murveit et al., 1993] Murveit, H., Butzberger, J., V., D., and M., W. (1993). Large-vocabulary dictation using SRI's DECIPHER speech recognition system: Progressive search techniques. In *Proc. ICASSP'93*, pages 319–322, Minneapolis, Minnesota, USA.

[Oerder and Ney, 1993] Oerder, M. and Ney, H. (1993). Word graphs: An efficient interface between continuous-speech recognition and language understanding. In *Proc. ICASSP'93*, pages 119–123, Minneapolis, Minnesota, USA.

[Pallett et al., 1995] Pallett, D., Fiscus, J., and Fisher, W. (1995). 1994 benchmark tests for the ARPA spoken language program. In *Proceedings of the ARPA Spoken Language Systems Technology Workshop*, Austin, Texas, USA. Advance Research Projects Agency, Morgan Kaufmann Publishers Inc.

[Paul and Baker, 1992] Paul, D. and Baker, J. (1992). The design for the Wall Street Journal-based CSR corpus. In *DARPA Workshop on Speech and Natural Language*, volume 11, pages 1–5, Harriman, New York, USA.

[Paulus and Lehning, 1994] Paulus, E. and Lehning, M. (1994). Die Evaluierung von Spracherkennungssystem in Deutschland. In *Fortschritte der Akustik, DAGA 94*, pages Teil A, 147–156. In German.

[Picone et al., 1990] Picone, J., Doddington, G. R., and Pallet, D. S. (1990). Phone-mediated word alignment for speech recognition evaluation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(3):559–562.

[Picone et al., 1986] Picone, J., Goudie-Marshall, K. M., Doddington, G. R., and Fisher, W. (1986). Automatic text alignment for speech system evaluation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):780–784.

[Price, 1990] Price, P. (1990). Evaluation of spoken language systems: The ATIS domain. In *DARPA Speech and Natural Language Workshop*, pages 91–95, Hidden Valley, California, USA.

[Price et al., 1988] Price, P., Fisher, W. M., Bernstein, J., and Pallett, D. S. (1988). The DARPA 1000-word Resource Management database for continuous speech recognition. In *Proc. ICASSP'88*, pages 651–654.

[Pye et al., 1995] Pye, D., Woodland, P., and Young, S. (1995). Large vocabulary multilingual speech recognition using HTK. In *Proc. EUROSPEECH'95*, volume 1, pages 181–184, Madrid, Spain.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

[Rabiner and Juang, 1986] Rabiner, L. R. and Juang, B.-H. (1986). An introduction to Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 3(1):4–16.

[Rabiner and Juang, 1993] Rabiner, L. R. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

[Reinecke, 1996] Reinecke, J. (1996). Evaluierung der signalnahen Spracherkennung im Verbundprojekt Verbmobil (Herbst 1996). Verbmobil Memo 113, Technische Universität Braunschweig, Braunschweig, Germany. In German. Available at `http://sbvsrv.ifn.ing.tu-bs.de/evaluierung`.

[Rosenfeld, 1994] Rosenfeld, R. (1994). *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, School of Computer Science, Carnegie Mellon University, Philadelphia, Pennsylvania, USA.

[Rosenfeld, 1995a] Rosenfeld, R. (1995a). The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation. In *Proceedings of the ARPA Spoken Language Technology Workshop*, pages 51–52, Austin, Texas, USA.

[Rosenfeld, 1995b] Rosenfeld, R. (1995b). An impact matrix for the 1994 CSR hub evaluation. In *Proceedings of the ARPA Spoken Language Technology Workshop*, pages 51–52, Austin, Texas, USA.

[Rosenfeld, 1995c] Rosenfeld, R. (1995c). Optimizing lexical and N-gram coverage via judicious use of linguistic data. In *Proc. EUROSPEECH'95*, pages 1763–1766, Madrid, Spain.

[Sakamoto and Matsunaga, 1995] Sakamoto, H. and Matsunaga, S. (1995). Detection of unknown words using garbage cluster models for continuous speech recognition. In *Proc. EUROSPEECH'95*, volume 2, pages 2103–2106, Madrid, Spain.

[Schaaf and Kemp, 1997] Schaaf, T. and Kemp, T. (1997). Confidence measures for spontaneous speech. In *Proc. ICASSP'97*, volume 2, page 875, Munich, Germany.

[Schultz and Rogina, 1995] Schultz, T. and Rogina, I. (1995). Acoustic and language modeling of human and nonhuman noises for human-to-human spontaneous speech recognition. In *Proc. ICASSP'95*, volume 1, page 293, Detroit, Michigan, USA.

[Schürmann, 1996] Schürmann, J. (1996). *Pattern Classification: A Unified View of Statistical and Neural Approaches.* Wiley-Interscience.

[Schwartz and Austin, 1991] Schwartz, R. and Austin, S. (1991). A comparison of several approximate algorithms for finding multiple (N-Best) sentence hypotheses. In *Proc. ICASSP'91*, pages 701–704.

[Shriberg, 1996] Shriberg, E. (1996). Disfluencies in Switchboard. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Shriberg, 1994] Shriberg, E. E. (1994). *Preliminaries to a Theory of Speech Disfuencies.* PhD thesis, University of California at Berkeley.

[Sloboda, 1995] Sloboda, T. (1995). Dictionary learning: Performance through consistency. In *Proc. ICASSP'95*, volume 1, page 453, Detroit, Michigan, USA.

[Sloboda and Waibel, 1996] Sloboda, T. and Waibel, A. (1996). Dictionary learning for spontaneous speech recognition. In *Proc. ICSLP'96*, Philadelphia, Pennsylvania, USA.

[Spiess, 1995] Spiess, M. (1995). A language model for compound words in speech recognition. In *Proc. EUROSPEECH'95*, volume 3, pages 1767–1770, Madrid, Spain.

[Suhm et al., 1993] Suhm, B., Woszcyna, M., and Waibel, A. (1993). Detection and transcription of new words. In *Proc. EUROSPEECH'93*, pages 2179–2182.

[Ueberla, 1994] Ueberla, J. (1994). *Analyzing and Improving Statistical Language Models for Speech Recognition.* PhD thesis, Simon Fraser University, Vancouver, British Columbia, Canada.

[Ueberla, 1995] Ueberla, J. (1995). An extended clustering algorithm for statistical language models. In *Proc. EUROSPEECH'95*, pages 1257–1260.

[Wahlster, 1993] Wahlster, W. (1993). Verbmobil—Translation of face-to-face dialogs. In *Proc. EUROSPEECH'93*, volume "Opening and Plenary Sessions", pages 29–38, Berlin, Germany.

[Weintraub, 1993] Weintraub, M. (1993). Keyword spotting using SRI's DECIPHER large-vocabulary speech-recognition system. In *Proc. ICASSP'93*, volume 2, pages 463–466, Minneapolis, Minnesota, USA.

[Weintraub, 1995] Weintraub, M. (1995). LVCSR log-likelihood ratio scoring for keyword spotting. In *Proc. ICASSP'95*, volume 1, page 297, Detroit, Michigan, USA.

[Wilpon et al., 1990] Wilpon, J., Rabiner, L., Lee, C.-H., and Goldman, E. (1990). Automatic recognition of keywords in unconstrained speech using Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(11).

[Yamada et al., 1994] Yamada, M., Komori, Y., and Yasunori, O. (1994). Active/non-active word control using garbage model—unknown word reevaluation in speech conversation—. In *Proc. ICSLP'94*, pages 823–826, Yokohama, Japan.

[Young, 1994] Young, S. R. (1994). Detecting misrecognitions and out-of-vocabulary words. In *Proc. ICASSP'94*, pages II–21–II–24, Adelaide, South Australia, Australia.

[Young and Ward, 1993] Young, S. R. and Ward, W. (1993). Learning new words from spontaneous speech. In *Proc. ICASSP'93*, pages II–590–II–591, Minneapolis, Minnesota, USA.