Karlsruhe Institute of Technology

Unsupervised Vocabulary Selection for Speech Recognition of Lectures

Diplomarbeit
von

Cand. Dipl.-Inf. Paul Märgner

am Institut für Anthropomatik
der Fakultät für Informatik

Erstgutachter:	Prof. Dr. Alex Waibel
Zweitgutachter:	Dr. Sebastian Stüker
Betreuender Mitarbeiter (CMU):	Prof. Dr. Ian Lane
Betreuender Mitarbeiter (KIT):	Dipl.-Inform. Kevin Kilgour

Bearbeitungszeit: 1. September 2011 – 29. Februar 2012

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 24. Februar 2012

Abstract

Recent advances in speech recognition technology have shown that automatic speech recognition systems can help to overcome language barriers in educational and scientific lectures. However, speech recognition of lectures involves many difficulties because the spoken words, such as technical terms and phrases, are very different between lectures of different topics. Therefore, speech recognition accuracy for new, previously unseen lectures is often poor. To overcome this problem, related work has used manually selected lecture-specific documents to adapt the system to the current lecture topic. Nevertheless, this manual approach is too expensive and time-consuming to allow effective speech recognition of all the diverse lectures at a university. Thus, in this work, an effective adaptation is introduced that solves this issue. Based on an initial text document that is available before the lecture begins, such as lecture slides, the proposed approach automatically adapts a speech recognition system without any human input. By automatically searching and downloading text documents from the world wide web, a document corpus is built, which is then used to adapt the speech recognition vocabulary and language model to the lecture topic. The adaptation is focused on the selection of a lecture-specific vocabulary by applying a novel vocabulary ranking scheme based on word features. In an experimental evaluation of this approach on six German lectures with different topics and speakers, the proposed approach showed significant improvements compared to a topic-independent baseline system. By using a lecture-specific vocabulary that was selected by applying the proposed method instead of a topic-independent baseline vocabulary, the out-of-vocabulary rate was relatively reduced on average by 53.0% per lecture. The adaptation of the language model to the lecture topic relatively improved the language model perplexity on average by 23.0% per lecture compared to a topic-independent baseline. Finally, the word error rate was relatively lowered on average by 12.5% per lecture by employing the lecture-specific vocabulary and language model compared to a topic-independent baseline system. These results show the effectiveness of the proposed approach.

Kurzfassung

Aktuelle Forschungsergebnisse haben gezeigt, dass automatische Spracherkennungssysteme dazu verwendet werden können, um Sprachbarrieren in Vorlesungen und Vorträgen zu überwinden. Allerdings beinhaltet die Spracherkennung von Vorlesungen viele Schwierigkeiten, da sich die verwendeten Wörter, wie z.B. Fachwörter, in Vorlesungen zu verschiedenen Themen stark unterscheiden. Daher ist die Spracherkennungsgenauigkeit bei neuen oder unbekannten Vorlesungen oft schlecht. Um dieses Problem zu lösen, werden Spracherkennungssysteme mit Hilfe von manuell ausgewählten, themenspezifischen Dokumenten adaptiert. Aber dieser manuelle Ansatz ist zu zeitaufwendig und teuer um gute Spracherkennung von allen unterschiedlichen Vorlesungen an einer Universität zu ermöglichen. Daher wird in dieser Arbeit ein neuer effektiver Adaptionsansatz zur Lösung dieses Problems vorgestellt. Das entwickelte Verfahren adaptiert ein Spracherkennungssystem automatisch an das Thema einer Vorlesung nur basierend auf einem initialen Textdokument, wie z.B. Vorlesungsfolien, ohne weitere Eingaben eines Menschen. Relevante Textdokumente werden automatisch im Internet gesucht und heruntergeladen. Anschließend wird die entstandene Dokumentensammlung für die Adaption des Spracherkennungsvokabulars und des Sprachmodells verwendet. Hierbei liegt der Fokus auf der Auswahl eines vorlesungsspezifischen Vokabulars mit Hilfe eines neuartigen Vokabularsortierungsverfahrens basierend auf Wortmerkmalen. In einer experimentellen Evaluation auf sechs deutschen Vorlesungen mit unterschiedlichen Themen und von unterschiedlichen Sprechern konnten mit Hilfe des vorgestellten Verfahrens deutliche Verbesserungen im Vergleich zu einem themenunabhängigen System erzielt werden. Die Verwendung eines vorlesungsspezifischen Vokabulars an Stelle eines themenunabhängigen Vokabulars verbesserte die Vokabularabdeckung, gemessen durch die sogenannte „out-of-vocabulary rate", relativ um durchschnittlich 53,0% pro Vorlesung. Die Adaption des Sprachmodells verringerte die Perplexität des Sprachmodells relativ um durchschnittlich 23,0% im Vergleich zu einem themenunabhängigen Sprachmodell. Abschließend wurde durch die Verwendung des vorlesungsspezifischen Vokabulars und Sprachmodells die Wortfehlerrate relativ um durchschnittlich 12,5% pro Vorlesung reduziert, verglichen mit einem themenunabhängigen System. Mit diesen Ergebnissen konnte die Effektivität des vorgestellten Ansatzes gezeigt werden.

Contents

1 Introduction 1

2 Theoretical Background 5
- 2.1 Automatic Speech Recognition . . . 5
  - 2.1.1 Vocabulary and Dictionary . . . 6
  - 2.1.2 Acoustic Model . . . 7
  - 2.1.3 Language Model . . . 8
  - 2.1.4 Summary and Evaluation . . . 11
- 2.2 Machine Translation . . . 12
- 2.3 Text Classification . . . 12

3 Related Work 15
- 3.1 Simultaneous Lecture Translation . . . 15
- 3.2 Adaptation Techniques . . . 17
- 3.3 Lecture-specific Adaptation . . . 20
- 3.4 Summary . . . 22

4 Problem Analysis 23

5 Concept 27
- 5.1 Document Collection . . . 28
- 5.2 Vocabulary Selection by using Feature-based Ranking . . . 29
- 5.3 Features for Vocabulary Selection . . . 30
  - 5.3.1 Document Features . . . 30
  - 5.3.2 Query Features . . . 31
  - 5.3.3 Word Features . . . 31
- 5.4 Lecture-specific Language Modeling . . . 34
- 5.5 Summary . . . 35

6 Implementation 37
- 6.1 Document Collection . . . 38
  - 6.1.1 Used Packages . . . 39
- 6.2 Vocabulary Selection and Language Model Adaptation . . . 41

7 Evaluation 43
- 7.1 Preparation . . . 44
- 7.2 Lecture Data . . . 44
- 7.3 Document Collection . . . 45
- 7.4 Vocabulary Selection . . . 46
  - 7.4.1 Boundaries . . . 46

7.4.2 Feature-based Vocabulary Selection 46
7.5 Lecture-dependent Language Model Adaptation 49
7.6 Lecture-dependent Speech Recognition 50
7.7 Discussion 51
7.8 Summary 52

8 Conclusion **53**

A Appendix **55**

Bibliography **63**

List of Tables

1.1 Examples of Lecture-specific Technical Terms and Names 2

4.1 Word Error Rate - Oracle Vocabulary 25

7.1 Lecture Data - Recording 44

7.2 Lecture Data - Lecture Slides 45

7.3 Document Collection Results (Up to 50 Results per Query) 45

7.4 Out-of-Vocabulary Rate - Topic-Independent Lecture Vocabulary and Google Book Unigrams 46

7.5 Out-of-Vocabulary Rate - DocCount, VocCount, and Doc+VocCount 48

7.6 Language Model Perplexity - 40k Vocabulary 50

7.7 Word Error Rate - Speaker Adaptation - 40k Vocabulary - "Search50"-Language Model 51

7.8 Percentage of English Words in the six German Test Lectures. 51

A.1 Extended Document Collection Results (up to 500 Results per Query) 55

A.2 Out-of-Vocabulary Rate - Baseline vocabularies 55

A.3 Out-of-Vocabulary Rate - Single Feature Score Ranking 56

A.4 Out-of-Vocabulary Rate - Baseline, Baseline+Slides, and Doc+VocCount 57

A.5 Language Model Perplexity - 90k Vocabulary 57

A.6 Language Model Perplexity - 300k Vocabulary 57

A.7 Word Error Rate - No Speaker Adaptation - 40k Vocabulary - "Search50"-Language Model 58

A.8 Word Error Rate - No Speaker Adaptation - 40k Vocabulary - "Search500"-Language Model 58

A.9 Word Error Rate - No Speaker Adaptation - 90k Vocabulary - "Search50"-Language Model 58

A.10 Word Error Rate - No Speaker Adaptation - 90k Vocabulary - "Search500"-Language Model 59

A.12 Word Error Rate - Speaker Adaptation - 40k Vocabulary - "Search50"-
Language Model 59
A.13 Word Error Rate - Speaker Adaptation - 40k Vocabulary - "Search50"-
Language Model 60
A.14 Word Error Rate - Speaker Adaptation - 40k Vocabulary - "Search500"-
Language Model 60
A.15 Word Error Rate - Speaker Adaptation - 90k Vocabulary - "Search50"-
Language Model 60
A.16 Word Error Rate - Speaker Adaptation - 90k Vocabulary - "Search500"-
Language Model 61
A.17 Word Error Rate - Speaker Adaptation - 300k Vocabulary - "Search50"-
Language Model 61
A.18 Word Error Rate - Speaker Adaptation - 300k Vocabulary - "Search500"-
Language Model 61

List of Figures

2.1 Components of a Speech Recognition System 5

3.1 The interACT Lecture Translation System with Head-up Display (left) and Targeted Audio Speakers (right) [WaF8] 15

3.2 Components of a Simultaneous Lecture Translation ystem 16

4.1 Out-of-Vocabulary Rate - 40k Baseline and 40k Baseline+Slides . . . 24

5.1 Vocabulary Selection and Language Model Adaptation 28

6.1 Vocabulary Selection and Language Model Adaptation - Implementation 37

6.2 Object Structure - Part of UML Diagram 39

6.3 Illustration of Parallel Processing in CorpusBuilder 40

7.1 Vocabulary Selection and Language Model Adaptation - Evaluation . 43

7.2 Average OOV Rate for all Features - 40k Vocabulary 47

7.3 Average OOV Rate of Baseline compared with Doc+VocCount 49

1. Introduction

Recent advances in streaming technologies allow research talks and lectures to be broadcast live from educational institutes around the world. This provides students with an unprecedented access to educational content no matter of their physical location. A recent example is Stanford's lecture "Introduction to Artificial Intelligence", which was offered to every student for free in the fall of 2011. More than 100,000 students from around the world signed up for this lecture. With this bold experiment, Stanford showed the potential that lies in distributed education. However, although physical barriers are reduced through web streaming technologies, language barriers remain. Lectures may be presented in a language the student cannot understand thus limiting the usefulness of such content. Similarly, due to the lack of subtitles, live audio-video content is unsuitable for the hearing impaired. To overcome these barriers, recent works have investigated both the use of speech-translation technologies to translate lectures in real-time [KWKN+08] and real-time lecture transcription for the hearing impaired [KaNA08]. Although it was shown in these papers that the methods are useful, the biggest downfall of these technologies is portability. The particular technologies rely on automatic speech recognition systems that are generally optimized to a specific lecture topic, as described in the next paragraphs.

Imagine a speech recognition system to be a human being who has just learnt a new language. The task of this person is to correctly write down every single word that will be said in this language. It is nearly impossible for this person to write down a word correctly that he has never seen or read before. To solve this problem for speech recognition systems, a word list is provided that contains all words that could be said. This list is called the speech recognition vocabulary. Creating this vocabulary is particularly difficult when dealing with a lecture of unknown topic. Every lecture has specific words, such as technical terms, which are common for the specific lecture topic but not for other lecture topics. Table 1.1 shows a list of examples of different topic-specific words of six different lecture topics. For example, in a chemistry lecture about polyamino carboxylic acids, the word "Ethylenediaminetetraacetic acid" might occur, but probably not in a math lecture. Another example: The name of "Jawaharlal Nehru", the first Prime Minister of independent India, is likely to occur in a history lecture about India, but not in a physics lecture. These

are only a few examples of topic-specific terms that can exist in lecture speech. A speech recognition vocabulary has to be specially adapted to correctly recognize these specific terms and create the correct lecture transcription.

Topic	Example group of words	Example
Chemistry	Different acids	Ethylenediaminetetraacetic acid
Biology	Different organisms	Family of Erethizontidae
History	Historic figures	Jawaharlal Nehru
Physics	Different particles	Higgsino
Arts	Artists, Names of paintings	Guernica by Pablo Picasso
Mathematics	Mathematical functions, symbols	Dirichlet eta function

Table 1.1: Examples of Lecture-specific Technical Terms and Names

If the system is not trained for the lecture topic, spoken words might be recognized as different words with a similar pronunciation. For example, if a system is not built for a specific physics lecture, it might not recognize words like "Newton" and "kinetics" correctly. When the lecturer says a sentence like "Today, I'm going to talk about Newton's second law, the kinetics of particles.", the system might recognize "Today, I'm going to talk about new tones second law, the genetics of particles." This sentence is not understandable anymore even though most of its words are recognized correctly. This example shows the importance of an adapted vocabulary.

Even more difficult to recognize correctly are *homophones*, which are words that have the same pronunciation but separate meanings [Brid06]. The only way to determine which word is meant is to use context information. For example, if the professor in a history lecture about ancient cities said: "Your homework for next week: Rome, the city.", it is clear that he is speaking about the Italian capital "Rome". However, imagine a social studies class in which the students have to conduct a survey in their local city. In this class, the phonetically identical sentence could have meant: "Your homework for next week: Roam the city." To recognize such a sentence correctly, specific information about the context, like topic of the current lecture, is needed. For the best performance, the system needs word sequence probabilities that enable the system to choose the correct word. In a speech recognition system, these probabilities are estimated by employing a statistical language model. The language model is trained by using sentences that are similar to the sentences that will be used in the lecture.

Therefore, for each new topic, significant effort and cost is required to manually transcribe similar lectures, without which the system will generally perform poorly. Only some work is already done for a more automatic adaptation process but these approaches are either ineffective in terms of vocabulary adaptation or they need an amount of human input that is too expensive for real world applications.

In the work at hand, an approach is developed that allows a speech recognition system to cope with the diverse educational and scientific lectures that are offered at a university, by automatically adapting itself to the lecture. The main part of this approach consists of an effective vocabulary selection method with as little human input as possible to allow a fast and cost effective solution.

The research was done at the Silicon Valley campus of Carnegie Mellon University, USA. During this work, a paper was published at the 13th Machine Translation Summit (MT Summit 2011) [MaLW11] and one at the 8th International Workshop on Spoken Language Translation (IWSLT 2011) [MKLW11]. Another paper was submitted and accepted to the 37th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012). The work was also presented and reviewed at the Carnegie Mellon Silicon Valley 2nd Annual Tech Showcase 2011 and received the "Best Project Award"[1] from an independent jury.

This thesis is structured as follows. Chapter 2 provides a short overview of the theoretical background and is followed by the related work, which is summarized in chapter 3. Chapter 4 analyzes the problem and presents exploratory experiments for a better understanding of the problem. In chapter 5, the concept of the implemented approach is given. The implemented software is described in chapter 6. Then, the approach is evaluated in chapter 7 on a set of six German lectures. The thesis ends with a conclusion in chapter 8.

[1] http://www.cmu.edu/silicon-valley/research/tech-showcase/index.html

2. Theoretical Background

This chapter provides an overview of the theoretical background of this work. The main techniques are automatic speech recognition (ASR), machine translation (MT), and text mining. The following introduction into ASR is mainly based on the book "Spoken Language Processing: A Guide to Theory, Algorithm and System Development" by Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon [HuAH01].

2.1 Automatic Speech Recognition

Automatic speech recognition (ASR) describes methods that convert speech into text. The components of a speech recognition system are illustrated in figure 2.1. In the first step (*signal processing*), the spectral features are extracted from the input speech waveform. The most commonly used spectral features for speech are *Mel-frequency cepstral coefficients* (MFCC). In [DaMe80], Davis and Mermelstein showed that the MFCC representation is beneficial for speech recognition. More details about MFCC can be found in [HuAH01]. Let $X = x_1, x_2, \ldots, x_N$ be the spectral feature representation sequence of the acoustic observations and $W = w_1, w_2, \ldots, w_M$ denote the corresponding word sequence. The *speech decoder* chooses the word sequence W that is the best match for the input speech representation X.

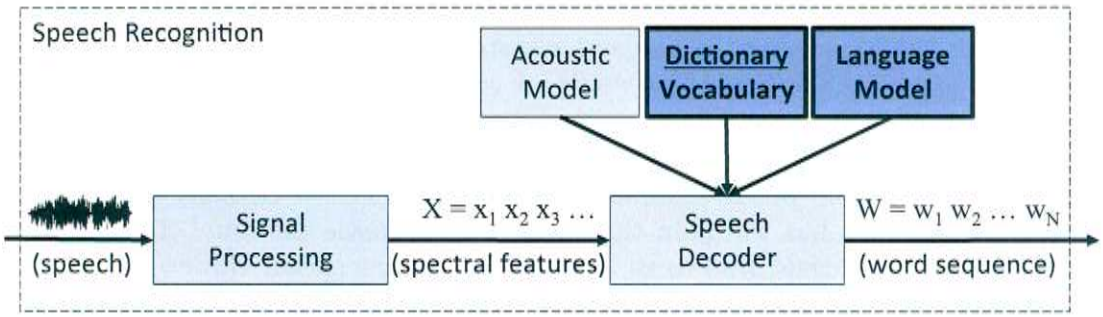

Figure 2.1: Components of a Speech Recognition System

Due to the high variability of speech, current ASR systems rely on statistical methods. These methods are based on the Bayes' theorem:

$$\mathrm{P}(W|X) = \frac{\mathrm{p}(X|W)\mathrm{P}(W)}{\mathrm{p}(X)} \tag{2.1}$$

The speech recognition decoder chooses the word sequence $\hat{W}$ that has the highest probability $\mathrm{P}(W|X)$ given the acoustic observation X. This means $\hat{W}$ is defined as:

$$\hat{W} = \arg\max_{W} \mathrm{P}(W|X)$$

By using the Bayes' theorem, we can write:

$$\hat{W} = \arg\max_{W} \mathrm{P}(W|X) = \arg\max_{W} \frac{\mathrm{p}(X|W)\mathrm{P}(W)}{\mathrm{p}(X)}$$

Since we are seeking the maximum, we do not have to divide the last term by $\mathrm{p}(X)$ because $\mathrm{p}(X)$ is common across all possible values for W. Consequently, the equation can be simplified:

$$\hat{W} = \arg\max_{W} \mathrm{P}(W|X) = \arg\max_{W} \mathrm{p}(X|W)\mathrm{P}(W) \tag{2.2}$$

where probability density $\mathrm{p}(X|W)$ is known as the *acoustic model* and the probability $\mathrm{P}(W)$ is called the *language model*. Equation 2.2 is called the "fundamental equation of speech recognition".

2.1.1 Vocabulary and Dictionary

The fundamental equation of speech recognition (eq. 2.2) is used to search across all possible word sequences $W = w_1, w_2, \ldots$ for the word sequences with the highest probability $\hat{W}$. To allow search across all possible word sequences, the number of possible words has to be limited. Thus, each word w_i is part of a predefined and limited vocabulary V with $w_i \in V$ and $|V| \in \mathbb{N}$. In vocabulary V, an inflected form is considered as a different word. The reason for that is that inflected forms usually have different pronunciations and usage patterns. Accordingly, the words "talk", "talks", "talked", and "talking" are counted as four different words in vocabulary V. Only words that are present in the active system vocabulary can be recognized by the ASR system because only these words are considered as possible word sequences. From this it follows that a larger vocabulary allows the ASR system to recognize more words. Nevertheless, a smaller vocabulary is preferred because this reduces the number of confusable words, leading to an improved speech recognition accuracy. In addition to that less words in the vocabulary decrease the number of possible word sequences and thus, lead to an increased processing speed. However, a smaller vocabulary obviously leads to a less flexible system. Furthermore, the user is usually not aware of which words the system can recognize. Spoken words that are not in the system's vocabulary are called *out-of-vocabulary words* (OOV words). In the best scenario, each OOV word leads to only one error in the ASR output. But

usually an OOV word also effects the speech recognition output of the surrounding words because the speech decoder tries to match the input sequence with words from the vocabulary. Thus, the number of OOV words directly effects speech recognition accuracy. Therefore, the *out-of-vocabulary rate* (OOV rate, eq. 2.3) is used to determine the quality of a vocabulary. Giving a text or transcript, the OOV rate is defined as follows:

$$\text{OOV rate} = \frac{\text{Number of OOV words in text}}{\text{Total number of words in text}} \cdot 100\% \tag{2.3}$$

The vocabulary is selected to achieve the lowest OOV rate given the available vocabulary size.

Since the system input is a spoken word and not a written word, a conversion from text to spoken word is needed. For this reason, the speech recognition dictionary creates this connection between words and pronunciations. For each word in the vocabulary, the dictionary contains one or more pronunciations, i.e. phoneme sequences. In [Asso99], a *phoneme* is defined as "the smallest segmental unit of sound employed to form meaningful contrasts between utterances in a language or dialect". By using phoneme sequences to represent words, the acoustic model only has to model the relationship between phonemes and spectral features. Therefore, it is independent from the vocabulary. The pronunciations are either determined manually or generated automatically by employing a text-to-speech (speech synthesis) system. An open-source system is the *Festival speech synthesis system*, which offers a general framework for building speech synthesis systems. The architecture of the Festival speech synthesis system is described by Paul Taylor et al. in [TaBC98]. The system is available at http://www.cstr.ed.ac.uk/projects/festival/.

2.1.2 Acoustic Model

In the search for the best word sequence W given the input speech signal X, the *acoustic model* provides the probability distribution $\mathrm{p}(X|W)$ of a speech signal given a word sequence. In modern ASR systems, the acoustic model usually models the relationship between small parts of the words, usually phonemes, and the input speech. The large number of words in modern ASR systems makes this approach more efficient. Additionally by focussing on phonemes, the acoustic model is independent of the vocabulary and the language model. Building an acoustic model is challenging since it is affected by the speaker, the microphone used, and the environment (small room, large hall, etc.) in which the recording took place. However, the acoustic model is not affected by the topic of the speech. The acoustic model is often built by applying *hidden Markov models* (HMM) and the probability density function is modeled by employing *Gaussian mixture models* (GMM). Each HMM models one phoneme. The model of a word is the concatenation of the HMMs of its phoneme sequence. More details on HMM and GMM, and their application in speech recognition can be found in [HuAH01]. As mentioned before, the acoustic model differs between different speakers and a new acoustic model training is needed to improve speech recognition. However, speaker adaptation can also be performed in an unsupervised manner. One unsupervised speaker adaptation approach is described in [LBGA+09].

2.1.3 Language Model

The recognition of the correct word sequence $\hat{W}$ is guided by the word sequence probability $P(W)$ provided by a statistical language model. The following description of language models is based on [ChGo98]. The language model provides the probability $P(W)$ over a given word sequence $W = w_1, w_2, \ldots, w_M$. This probability reflects how often the word sequence W occurs in a language. The most common language models are n-gram models. The most simple n-gram model is called *unigram* (1-gram) model. It provides the probability of a single word without taking any previous words into account, $P(w_i)$, *bigrams* (2-grams) models give the probability of a word given the previous word, $P(w_i|w_{i-1})$, and *trigrams* (3-grams) models give the probability of a word given the previous two words, $P(w_i|w_{i-2}, w_{i-1})$. In general, an n-gram model provides the probability of a word given the $n-1$ previous words: $P(w_i|w_{i-(n-1)}, \ldots, w_{i-1})$. The idea of a language model based on an n-gram model is that the probability $P(W)$ of a word sequence W composed of the words $w_1 \ldots w_M$ can be expressed as:

$$P(W) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\ldots P(w_M|w_1\ldots w_{M-1}) = \prod_{i=1}^{M} P(w_i|w_1\ldots w_{i-1})$$

We can approximate the probability $P(W)$ by employing n-gram models if we assume that the probability of a word depends only on the $n-1$ immediately preceding words. For example, we can use a bigram model to approximate $P(W)$ if we make the assumption that the probability of a word depends only on the immediately preceding word. With this assumption $P(W)$ can be expressed as:

$$P(W) = \prod_{i=1}^{M} P(w_i|w_1\ldots w_{i-1}) \approx \prod_{i=1}^{M} P(w_i|w_{i-1})$$

The n-gram probabilities are usually estimated by using a text corpus, simply counting the n-gram occurrences, and normalizing them. Let $c(w_{i-1}w_i)$ be the number of times the bigram $w_{i-1}w_i$ occurs in the given text corpus. Then, we can take:

$$P_{ML}(w_i|w_{i-1}) = \frac{c(w_{i-1}w_i)}{\sum_{w_i} c(w_{i-1}w_i)}$$

This estimate is called the *maximum likelihood* (ML) estimate of $P(w_i|w_{i-1})$

Now, consider the following small example. Let the given text corpus be composed of three sentences ("MARY WALKED AROUND", "TIM WALKED TO PETER", "SHE WALKED TO HIM") and let us calculate P(TIM WALKED AROUND) by applying a bigram language model. Let "<START>" mark the start of the sentence and let "<END>" indicate the end of the sentence. We have the following bigram probabilities:

$$\mathrm{P}(\textsc{Tim}|\textsc{<start>}) = \frac{c(\textsc{<start> Tim})}{\sum_w c(\textsc{<start>}\ w)} = \frac{1}{3}$$

$$\mathrm{P}(\textsc{walked}|\textsc{Tim}) = \frac{c(\textsc{Tim walked})}{\sum_w c(\textsc{Tim}\ w)} = \frac{1}{1}$$

$$\mathrm{P}(\textsc{around}|\textsc{walked}) = \frac{c(\textsc{walked around})}{\sum_w c(\textsc{walked}\ w)} = \frac{1}{3}$$

$$\mathrm{P}(\textsc{<end>}|\textsc{around}) = \frac{c(\textsc{around <end>})}{\sum_w c(\textsc{around}\ w)} = \frac{1}{1}$$

This gives us

$$\begin{aligned}&\mathrm{P}(\textsc{Tim walked around})\\ &= \mathrm{P}(\textsc{Tim}|\textsc{<start>}) \cdot \mathrm{P}(\textsc{walked}|\textsc{Tim}) \cdot \mathrm{P}(\textsc{around}|\textsc{walked}) \cdot \mathrm{P}(\textsc{<end>}|\textsc{around})\\ &= \frac{1}{3} \cdot \frac{1}{1} \cdot \frac{1}{3} \cdot \frac{1}{1} = \frac{1}{9}\end{aligned}$$

Smoothing

In practice, the given text data is often sparse. A problem occurs in the current approach, when we observe a previously unseen bigram. If we calculate the probability of the sentence Peter walked around. We get

$$\mathrm{P}(\textsc{walked}|\textsc{Peter}) = \frac{c(\textsc{Peter walked})}{\sum_w c(\textsc{Peter}\ w)} = \frac{0}{1} = 0$$

and therefore, P(Peter walked around) = 0. This means that the probability of P(Peter walked around|X) given any acoustic input X will be zero. Thus, the speech recognition system will never recognize this sentence. Obviously, this is undesired. There is at least some probability that this sentence occurs.

This problem is addressed by smoothing. In [ChGo98], Chen and Goodman provide a good description of what smoothing is: *"The term smoothing describes techniques for adjusting the maximum likelihood estimate of probabilities to produce more accurate probabilities. The name smoothing comes from the fact that these techniques tend to make distributions more uniform, by adjusting low probabilities such as zero probabilities upward, and high probabilities downward. Not only do smoothing methods generally prevent zero probabilities, but they also attempt to improve the accuracy of the model as whole. Whenever a probability is estimated from few counts, smoothing has the potential to significantly improve estimation."*

A simple smoothing method is *absolute discounting* (AD) described in [NeEK94]. The idea of this smoothing technique is to reduce the probability of n-grams that did occur in the text corpus by a fixed value. The now free probability mass is

$$P_{AD}(\text{A}|\text{THAT}) = P_{AD}(\text{THOU}|\text{THAT})$$

However, it seems that we should have

$$P(\text{A}|\text{THAT}) > P(\text{THOU}|\text{THAT})$$

because the word A is much more common than the word THOU. This problem is engaged by *Jelinek-Mercer smoothing* (JM) [JeMe80]. Their approach is to linearly interpolate higher-order n-gram models with lower-order n-gram models. For the example above, this means to interpolate the bigram model and the unigram model with the following equation:

$$P_{JM}(w_i|w_{i-1}) = \lambda \cdot P_{ML}(w_i|w_{i-1}) + (1 - \lambda) \cdot P_{ML}(w_i)$$

where $0 \leq \lambda \leq 1$. Because $P_{ML}(\text{A}|\text{THAT}) = P_{ML}(\text{THOU}|\text{THAT}) = 0$ and if $P_{ML}(\text{A}) > P_{ML}(\text{THOU})$, we get

$$P_{JM}(\text{A}|\text{THAT}) > P_{JM}(\text{THOU}|\text{THAT})$$

as desired.

One of the best smoothing techniques was proposed by Kneser and Ney in [KnNe95]. To introduce their approach, let us now consider a small example. Imagine a training corpus in which a word, say FRANCISCO, occurs often but it only occurs after a single word, say SAN. This means that the unigram probability $P_{ML}(\text{FRANCISCO})$ will be high because the $c(\text{FRANCISCO})$ is high. Therefore, a smoothing approach like Jelinek-Mercer would assign a relatively high probability to the word FRANCISCO occurring after any different words in a previously unseen bigram. However, given the training data, it seems to be unlikely that FRANCISCO occurs after a different word because it occurs only after SAN in the training data. Kneser and Ney's idea was that the unigram probability should not be proportional to the occurrence counts of a word, but instead to the number of different words it follows. Chen and Goodman have shown in [ChGo98] that *Kneser-Ney smoothing* (KN) works well and outperforms most other smoothing techniques. More details of the introduced smoothing approaches can be found in [ChGo98].

Interpolation

When dealing with different topics, it can be useful to combine language models from different sources. The most common approach is linear interpolation. In [Rose96], linear language model interpolation is defined as follows:

Given k language models $\{\mathrm{P}_i(w|h)\}_{i=1,\ldots,k}$ (h is the word history, for example the two previous words for a trigram model), we can combine them linearly with:

$$\mathrm{P}_{\text{Interpolated}}(w|h) = \sum_{i=1}^{k} \lambda_i \mathrm{P}_i(w|h) \tag{2.4}$$

where $0 \leq \lambda_i \leq 1$ and $\sum_i \lambda_i = 1$.

Evaluation

The quality of a language model can be determined by calculating the language model perplexity. The perplexity of a language model measures how unexpected a given text is. This means the smaller the perplexity, the better the language model fits the given text. In [HuAH01], the perplexity $\mathrm{PP}(W)$ of a language model $\mathrm{P}(W)$ is defined as follows:

$$\mathrm{PP}(W) = 2^{-\frac{1}{N_W}\log_2 \mathrm{P}(W)} \tag{2.5}$$

where N_W is the length of the text W measured in words.

2.1.4 Summary and Evaluation

Automatic speech recognition is used to convert an input speech signal into text. This is realized by performing search across all possible word sequences. First, the input speech waveform is converted into a spectral feature representation. Then, the word sequence is selected that has the highest probability given a spectral feature input sequence. This probability is determined by applying the "fundamental equation of speech recognition" (Eq. 2.2). The application of the fundamental equation relies on a vocabulary/dictionary, a acoustic model, and a language model. The vocabulary provides the list of all possible words and the dictionary contains phoneme sequences for all words in the vocabulary (section 2.1.1). The acoustic model is used to determine the probability of a spectral feature sequence given a phoneme sequence (section 2.1.2). And, the language model offers the probability that a word sequence occurs in language (section 2.1.3). By using these probabilities, the ASR system calculates the most likely output word sequence.

The output of a ASR system is evaluated by calculating the word error rate. In [HuAH01], the *word error rate* (WER) is defined as follows:

$$\mathrm{WER} = \frac{\mathrm{Subs} + \mathrm{Dels} + \mathrm{Ins}}{\textit{Number of words in correct sentence}} \cdot 100\% \tag{2.6}$$

where (Subs + Dels + Ins) stands for the minimum sum of the following three types of word recognition errors:

- *Substitution*: An incorrect word was substituted for the correct word.
- *Deletion*: A correct word was omitted in the recognized sentence.
- *Insertion*: An extra word was added in the recognized sentence.

2.2 Machine Translation

Statistical machine translation (MT) component translates a given text W_A in language A into a text W_B in language B. This translation is generated by determining which string W_B maximizes the probability $\mathrm{p}(W_B|W_A)$ that the string W_B is the translation of the given string W_A. The probability is approximated by using a *translation model*, which provides the probability $\mathrm{p}(W_A|W_B)$ that a string W_A in language A is the translation of the string W_B in language B, and a *language model* for language B (section 2.1.3), which offers the probability $\mathrm{p}(W_B)$ that the string W_B occurs in the language B. By applying the Bayes' theorem (eq. 2.1), $\mathrm{p}(W_B|W_A)$ can be expressed as:

$$\mathrm{p}(W_B|W_A) \propto \mathrm{p}(W_A|W_B)\mathrm{p}(W_B)$$

By using this relation, the optimal translation $\hat{W}_B$ can be expressed as:

$$\hat{W}_B = \arg\max_{W_B} \mathrm{p}(W_B|W_A) = \arg\max_{W_B} \mathrm{p}(W_A|W_B)\mathrm{p}(W_B)$$

The translation model is trained by employing parallel corpora, which contain multiple sentences in language A and their translation into language B. Initially, translation models were word-based [VoNT96]. However, in [KoOM03], Koehn et al. showed improvements in translation quality by applying phrase-based translation models. More details about statistical machine translation can be found in [Koeh10].

2.3 Text Classification

This section introduces measures to weight words and compare documents. These measures have been used successfully for text classification in several text mining and information retrieval tasks.

Term frequency–inverse document frequency

The *TF-IDF measure* (term frequency–inverse document frequency) is a well-known measure to determine how important a word is to a document in a document corpus. The main idea of TF-IDF was introduced by Karen Spärck Jones in 1972; the paper was reprinted in the year 2004 in [Jone04]. She discovered that words that occur in many documents are usually less specific and, thus, often not important to a specific document. Based on this idea, the *inverse document frequency* (IDF) is used as a weighting factor [SaBu88]. TF-IDF is defined as follows:

$$\mathrm{idf}_w = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing word } w}\right) \tag{2.7}$$

$$\mathrm{tfidf}_{d,w} = \mathrm{tf}_{d,w} \cdot \mathrm{idf}_w \tag{2.8}$$

where the term frequency $\mathrm{tf}_{d,w}$ is defined as the frequency of the word w in the document d.

Google Book n-gram dataset

To calculate IDF, a document corpus is needed that contains documents on different topics. A large document corpus is the *Google Book n-gram dataset*. It can be downloaded for free at http://books.google.com/ngrams/datasets. This corpus contains n-gram counts (up to 5-gram) for all digitalized books in Google Books in July 2009. For each n-gram, the dataset contains how often the word occurs in total (*match count*), on how many pages (*page count*), and in how many books (*volume count*). These numbers are listed by the year the book was published. The Google Book n-gram data set is available for multiple languages. The data collection is described by Jean-Baptiste Michel et al. in [MiSA11].

TF-IDF vector

The *TF-IDF vector* for any document d is defined as follows:

$$\mathbf{tfidf}_d = \begin{pmatrix} \text{tfidf}_{d,w_1} & \dots & \text{tfidf}_{d,w_n} \end{pmatrix}^T \tag{2.9}$$

where $w_1, ..., w_n$ are all unique words occurring in a predefined vocabulary, tfidf_{d,w_i} is the TF-IDF measure of the word w_i given the document d.

Word frequency vector

The *word frequency vector* for any document d is defined as follows:

$$\mathbf{freq}_d = \begin{pmatrix} \text{c}_d(w_1) & \dots & \text{c}_d(w_n) \end{pmatrix}^T \tag{2.10}$$

where $w_1, ..., w_n$ are all unique words occurring in a predefined vocabulary, $\text{c}_d(w_i)$ is the number of occurrences of the word w_i in document d.

Cosine similarity

The *cosine similarity* (equation 2.11) is a similarity measure, which has been found to be effective in information retrieval. The cosine similarity calculates the cosine distance between two vectors $\mathbf{a} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix}^T$ and $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \dots & b_n \end{pmatrix}^T$ in the following manner:

$$\text{cosine}(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} (a_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (b_i)^2}} \tag{2.11}$$

3. Related Work

This chapter provides an overview of related work. First, the interACT simultaneous lecture translation system is described. Then, several vocabulary selection techniques are described, followed by an introduction of language model adaptation and translation model adaptation approaches. In the last part of this chapter, several papers on lecture-specific adaptation are discussed.

3.1 Simultaneous Lecture Translation

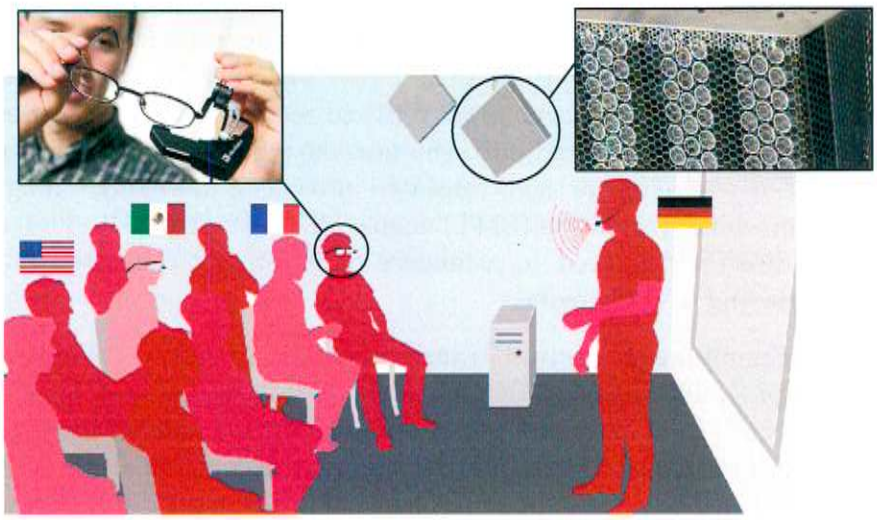

Figure 3.1: The interACT Lecture Translation System with Head-up Display (left) and Targeted Audio Speakers (right) [WaF8]

The approach introduced in this thesis will be evaluated on the interACT simultaneous lecture translation system. The *interACT simultaneous lecture translation system* is a real-time lecture translation system developed at the International Center for Advanced Communication Technologies (interACT) at Karlsruhe Institute of Technology, Germany, and Carnegie Mellon University, USA. A detailed description

of the system and its components can be found in [F07]. This lecture translation system, illustrated in figure 3.1, simultaneously translates lectures in real-time from the speaker's language into multiple languages required by the audience. To minimize the distraction to the audience, the system delivers translation as either text or speech output. The translated text is displayed either on screens in the lecture room, on a website accessible on mobile devices or on head-up displays. These technologies are especially useful for listeners who have partial knowledge of the speaker's language and want to have supplemental language assistance. Spoken translation output can be listened to either via headphones or targeted audio speakers [OlPL05], which make it possible to send the translated audio stream only to a small group of people while the other listeners are not disturbed.

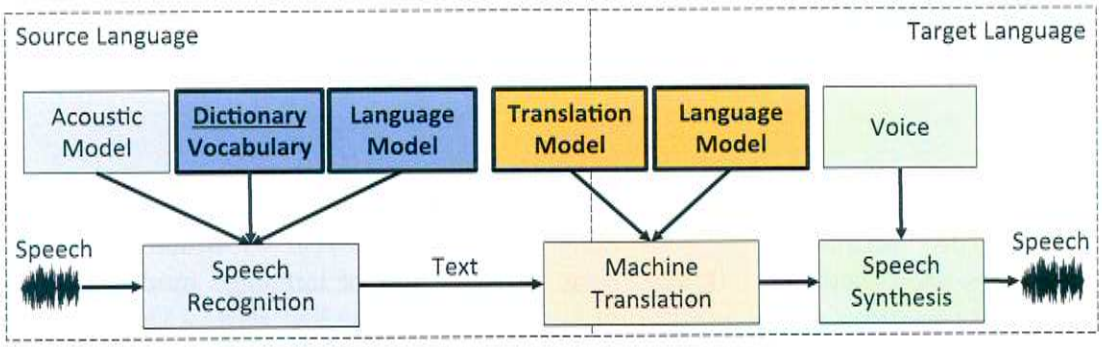

Figure 3.2: Components of a Simultaneous Lecture Translation ystem

Figure 3.2 illustrates the three main components of this lecture translation system: Automatic speech recognition (ASR, section 2.1), machine translation (MT, section 2.2), and speech synthesis (Text-to-Speech, TTS). They used the *Janus Recognition Toolkit* (JRTk) as ASR component [SMFW01] to recognize the input speech. The resulting text output is segmented into sentence-like units, which are then passed to MT. The resulting segments are then translated into one or more target languages via the statistical machine translation (SMT) engine STTK [VZHT+03]. The translated text is either directly displayed to attendees or optionally converted into speech output by employing a TTS engine.

The interACT Simultaneous Lecture Translation System relies on topic-independent models. The topic-independent system vocabulary was selected based on word occurrence counts in both in-domain and out-of-domain corpora and lecture-independent models for speech recognition were built by using these corpora. In [F07], Christian Fügen et al. identified topic-specific words (special terms, named entities, special expressions) as one of the main issues in lecture speech recognition and lecture speech translation. Muntsin Kolss et al. also encountered this problem in [KWKN+08]. They improved their system by adding manually selected topic-specific data to the vocabulary. This approach improved their OOV rate from 3.1% to 2.3% (25.8% relative improvement).

Topic Adaptation

As illustrated in figure 3.2, the simultaneous lecture translation system relies on three subsystems, automatic speech recognition (ASR), machine translation (MT), and speech synthesis (text-to-speech, TTS). These subsystems need several different models. The ASR system uses three models, an acoustic model (section 2.1.2), which models the phonetic units in the input speech, a recognition dictionary (section 2.1.1), which contains the pronunciation of all individual words in the recognition vocabulary, and a language model (LM, section 2.1.3), which provides the likelihood of word sequences. The MT system (section 2.2) needs a translation model, which models the likelihood of translations from the source language to the target language, and a language model of the target language. The TTS system needs a pronunciation dictionary for the target language. The question arises, which of these models need to be adapted when the lecture translation system is applied on a different lecture topic?

As described in section 2.1.2, the acoustic model is affected by the speaker, by the microphone used, and by the environment in which the recording took place, but not by the topic. Therefore, it is not necessary to adapt the acoustic model to the lecture topic. Differing from the acoustic model, the recognition vocabulary has to be adapted to include the different scientific terms depending on the lecture topic (see examples in chapter 1). When a new vocabulary is selected, the recognition dictionary can be built by using automatic approaches to create the pronunciations for all words in the vocabulary (see section 2.1.1). Additionally to the vocabulary, the probability of word sequences change with the lecture topic (see examples in chapter 1 and in section 2.1.3). Hence, the LM has to be adapted to the lecture topic. The LM adaptation data should contain enough text to cover the adapted vocabulary. Since a new topic means new vocabulary, translations have to be provided for these new words, too. Consequently, the translation model has to be adapted. Lastly, the target language model needs to be adapted to cover the new vocabulary of the new topic in the target language. The TTS system does not have to be adapted to the topic because most of these systems already perform well enough on any text input. To conclude, vocabulary, language model, and translation model have to be adapted to the lecture topic. This leaves the question, what is needed to adapt these models and how can it be done?

3.2 Adaptation Techniques

Vocabulary Selection

The quality of a vocabulary depends on its OOV rate while not exceeding its limited size, as described in section 2.1.1. The goal is a low OOV rate and a low vocabulary size. It is clear that, in general, a smaller vocabulary means a higher OOV rate. Therefore, some vocabulary selection technique is needed. The selection of a vocabulary is generally performed in two steps. First, a text corpus is chosen and then, a vocabulary is selected from this corpus. The vocabulary selection method depends on the quality of the text corpus. As described in [HuAH01], if the text corpus has the same word distribution as the spoken words, the minimum OOV rate vocabulary is selected by choosing the most frequent words of the text corpus for the vocabulary.

In [KWKN+08], the vocabulary coverage was improved by simply adding topic-specific words to the vocabulary. The words were chosen with the knowledge of which words are missing in current vocabulary. In a realistic scenario, this knowledge is usually not available before the lecture has started and can only be used for a second pass.

A comparison of four different vocabulary selection techniques was performed by Venkataraman et al. in [VeWa03]. All techniques depended on a training corpus that contains multiple documents. Additionally, one document is needed that has to be a partial observation of the actual transcript. They called it partially visible *held-out data*. In their evaluation, they tried to select an optimal vocabulary for English broadcast news and they used a 3 hour transcript of broadcast news as held-out data. All four techniques selected the vocabulary based on the estimated real counts x_i of each word w_i. Venkataraman et al. defined x_i by using the function Φ_i as follows:

$$x_i = \Phi_i(n_{i,1}, \ldots, n_{i,m}) = \sum_j \lambda_j n_{i,j} \tag{3.1}$$

where $n_{i,j}$ are the normalized counts of the word i in document j, m is the total number of documents, and $0 \leq \lambda_j \leq 1$ is a weight for each document j. Each vocabulary selection method used a different approach to choose the document weights λ_j.

- **Uniform:** Their baseline method assigned each document j the same weight λ_j:

$$\forall j \quad \lambda_j = \frac{1}{m}$$

- **Maximum likelihood count estimation:** This technique used the normalized counts $n_{i,j}$ as probability estimates of the word w_i given the document j. Formally, $\mathrm{P}(w_i|j) = n_{i,j}$. The document weights λ_j are determined by optimizing the following equation:

$$\hat{\lambda}_1, \ldots, \hat{\lambda}_m = \arg\max_{\lambda_1,\ldots,\lambda_m} \prod_{i=1}^{|V|} \Big(\sum_j \lambda_j \mathrm{P}(w_i|j) \Big)^{C(w_i)}$$

  where $C(w_i)$ is the count of w_i in the partially observed held-out data and V is the set of all words.

- **Document-similarity based on Euclidean distance:** In this approach, the document weights λ_j were chosen based on the Euclidean distance between the document and the held-out data. Formally, they defined:

$$D_{Euclid,j} = \sqrt{\sum_{i=1}^{|V|} \left(n_{i,0} - n_{i,j}\right)^2}$$

  where $n_{i,0}$ are the normalized counts of w_i in the held-out data. The document weights were defined as:

$$\lambda_j = \frac{1/D_{Euclid,j}}{\sum_k \left(1/D_{Euclid,k}\right)}$$

- **Document-similarity based on Kullback-Leibler divergence:** A similar approach is based on the Kullback-Leibler (KL) divergence [KuLe51]. The distance between the held-out data and a document was defined as follows:

$$D_{KL,j} = \sum_{i=1}^{|V|} n_{i,0} \log_2 \left(\frac{n_{i,0}}{n_{i,j}} \right)$$

Based on $D_{KL,j}$, the document weights were defined as:

$$\lambda_j = \frac{1/D_{KL,j}}{\sum_k (1/D_{KL,k})}$$

In the evaluation presented in [VeWa03], the maximum likelihood count estimation achieved the lowest OOV rate. The performance of the approach based on the Euclidean distance was about the same as the approach with uniform weights. The worst results were obtained by the approach based on the Kullback-Leibler divergence. These results showed the advantage of the maximum likelihood approach. However, this approach needs data, such as a transcript of a similar lecture, to estimate the real counts. This data might not be available.

Language Model

Similar to vocabulary selection, the language model adaptation is based on a text corpus (see 2.1.3) that provides topic specific data. In [KaNA08], this data is used to train a language model. This language model is than interpolated by using the previous language model with an approach like the one described in section 2.1.3 eq. 2.4. Multiple text corpora were used in [YISF+07] and interpolated with different weights with the previous language model. The weights were optimized manually to achieve the best results. In [MaTN08], the weights were determined based on a maximum likelihood estimation. Similar to the maximum likelihood estimation for vocabulary selection, this approach needs text data, such as a transcript of a similar lecture, to perform this estimation.

Translation Model

Translation models are usually trained by using parallel corpora that contain a text in language A and the translation of this text in language B. These corpora are usually scarcely available especially when dealing with many different topics. However, A. Eisele and J. Xu have developed a method to improve translation quality by employing comparable corpora [EiXu10]. For this approach, a preliminary machine translation (MT) model and two corpora in two different languages on the same topic are needed. The preliminary MT models are used to identify parallel parts in these two corpora. The parallel parts are used to improve the MT model. These two steps are repeated in a bootstrapping loop to improve the MT model further. Since two topic-specific corpora are needed anyway to adapt the two language models in source and target language, this approach seems to be useful to adapt the translation model.

3.3 Lecture-specific Adaptation

Recent Progress in the MIT Spoken Lecture Processing Project

In [GHHW04], Glass et al. analyzed lecture speech and found out that the lecture vocabulary can be considered as a combination of a topic-independent and a topic-specific vocabulary. The topic-independent vocabulary consists of words common in lecture speech, and is similar to words used in conversational dialogues. In contrast to that, the topic-specific vocabulary contains words that are specific to the topic, such as technical terms. These words are usually highly relevant, thus it is important to recognize them. However, these words are rarely used in common domains usually used in speech recognition, such as broadcast news. Nevertheless, these topic-specific words uttered in lectures are often the same topic-specific words written in related text documents. When dealing with different lectures, the topic-independent vocabulary has to be selected only once, while the topic-specific vocabulary usually changes between lectures. Within the MIT Spoken Lecture Processing Project [GHCM+07], Glass et al. performed lecture adaptation by leveraging from any supplemental text material that is available prior to the lecture, including lecture slides, journal articles, and book chapters. They used one topic-independent vocabulary and then added any vocabulary from supplemental text material to it to create a topic-specific vocabulary. The language model was adapted by interpolating a topic-independent language model with a language model built by using the supplemental text material. This interpolation was implemented by employing the SRI Language Modeling Toolkit. By applying these approaches, Glass et al. were able to lower the *out-of-vocabulary rate* (OOV rate) from 1.03% to 0.64% and reducing the *word error rate* (WER) from 33.6% to 31.3%. The WER was further reduced to 28.4% by applying unsupervised acoustic model adaptation. Although this approach improved vocabulary coverage and speech recognition accuracy, it relied on many manually provided text documents. These documents have to contain all topic-specific vocabulary and offer enough content to adapt the language model.

Dynamic Language Model Adaptation by Using Presentation Slides

Yamazaki et al. introduced an approach for joint vocabulary and language model adaptation in [YISF+07] to aid the archiving and search of lectures. They focused their work on language model adaptation based on the lecture slides. Their adaptation is performed in three steps: vocabulary adaptation, global language model adaptation, and then local language model adaptation. First, they added missing words from the lecture slides to the active system vocabulary. No vocabulary selection was performed. Second, they adapted their baseline language model by using the text on the lecture slides (global adaptation). In detail, they adapted the n-gram counts by adding the weighted n-gram counts of the slides to the n-gram count of the baseline training corpus. For each n-gram N_i, the globally adapted frequency $c_G(N_i)$ was calculated as follows:

$$c_G(N_i) = c_B(N_i) + \omega_1 c_A(N_i)$$

where $c_B(N_i)$ is the frequency of n-gram N_i that appear in the baseline training data, $c_A(N_i)$ is the frequency of the n-gram N_i that appear in all slides, and ω_1 is a weight coefficient. They optimized ω_1 experimentally. In the last step, they adapted

the language model locally to the individual slide that is currently shown in the lecture. They calculated the locally adapted frequency $c_L(N_i)$ as follows:

$$c_L(N_i) = c_G(N_i) + \omega_2 c_C(N_i)$$

where $c_C(N_i)$ is the frequency of n-gram N_i in the current slide, and ω_2 is a weight coefficient. They constructed a new language model for each slide by using the frequency $c_L(N_i)$ for all n-grams. They applied the Witten-Bell method [WiBe91] for back-off smoothing. They evaluated their approach on four Japanese lectures. Their results showed improvements compared to a baseline system. The OOV rate was reduced from 4.3% to 3.4%. The word error rate was reduced by 3.0% (absolute values were not provided by the authors).

Automatic Lecture Transcription by Exploiting Presentation Slides

A similar approach was used by Kawahara et al. in [KaNA08]. They applied the approach for automatic subtitling of lectures for the hearing impaired in [Kawa10]. For vocabulary adaptation, all out-of-vocabulary words that appeared in the slides were added to the active recognition vocabulary. This brought a small improvement in WER of 0.2% absolute[1]. For language model adaptation, Kawahara et al. compared two global approaches. The first approach used PLSA (Probabilistic Latent Semantic Analysis) to select documents similar to the lecture slides from a large corpus of lecture transcripts. These text documents were then used to adapt the language model. The second approach adapted the language model integrating related web text. They selected three keywords based on the tf-idf measure from each slide, and then used these keywords as one search query. They evaluated these two approaches on two different kinds of lectures (computer science course and automatic speech recognition tutorial). Both approaches improved word accuracy. The best results on the CS lecture were achieved with web text adaptation. The word error rate improved from 41.20% (baseline) to 39.50% (web text) (40.59% with PLSA). The PLSA approach performed better on the ASR tutorial. In this case, the word error rate improved from 28.17% (baseline) to 27.60% (PLSA) (27.63% with web text). Their explanation was that the topic of the CS lecture was not well covered in the corpus used for the PLSA approach, while the topic of the ASR tutorial was covered. In an additional evaluation, they applied local adaptation by applying a cache model additionally to the PLSA adaptation. In a cache model, the probabilities of preceding words (cache) that were recognized during ASR is heightened in the language model, assuming that they are more likely to be re-used. Additionally, they extended the scheme of the cache model by including the current slide's words to the cache. By employing the cache model, they were able to improve word error rate further to 39.03% for the CS lecture and 26.89% for the ASR tutorial.

Web-based Language Modeling for Automatic Lecture Transcription

A different approach for language model adaptation was introduced by Munteanu et al. [MuPB07] to improve the search for lectures in a lecture archive. They built a new language model entirely based on documents available in the world wide web (hereafter referred to as web). Their method was to search the web for

[1] Kawahara et al. did not published the OOV rate improvement in absolute numbers.

PDF documents by using the text from the lecture slides. They assumed that the lecture slides were mostly organized in bullet form (one idea constitutes a line on a slide). Thus, they used every single line in the lecture slides as one search query even when the line is part of a larger text. They limited the search to PDF documents in English language. For each of their four test lectures, they collected three different text corpora by searching for 10, 20, or 30 documents per query. Next, they performed corpus filtering by comparing each line in the corpus with two initial dictionaries. Every line with at least four words and more than one non-dictionary word was removed. As initial dictionaries they used the 5k-word WSJ dictionary, which is included in the SONIC toolkit, and the 100K-word CMU pronunciation dictionary, for further details see [MuPB07]. All remaining words in the corpus were used as vocabulary. For language model training they used only these web corpora and a vocabulary size of 40k words. The average OOV rate on their four English lectures were on average 0.3% for the baseline and below 0.1% for all other models. They did not use specific vocabulary selection techniques. They evaluated their system on four lectures of the same course by the same lecturer. They used three different scopes for the language models of each lecture: Adaptation on all lecture slides, adaptation on each individual slide, and adaptation on a cluster of slides. This approach improved transcription accuracy compared to a lecture-independent baseline. The best results were received by adapting on all lecture slides. The word error rate (WER) dropped on average from 48.63% (baseline) to 43.54% (20 docs per query) and 43.43% (30 docs per query). The WER results were very similar for the different numbers of documents per queries. For two lectures 20 documents per query received the best results, for the other two lectures the best results were received by using 30 documents per query. On average, 30 documents per query achieved the lowest WER.

3.4 Summary

Related work showed the opportunities for modern education that lie in speech recognition technologies. Especially, the interACT simultaneous lecture translation system, which simultaneously translates lecture speech into multiple languages, has the potential to break down language barriers. Nevertheless, the interACT lecture translation system has difficulties dealing with different lecture topics, such as topic-specific terms. Recent works showed improvements in speech recognition accuracy by applying approaches for topic adaptation. However, these approaches were mainly focused on the adaptation of the language model, the improvement of the vocabulary was barely addressed. Furthermore, most approaches relied on manually selected data that might not be available. In addition to that, the manual selection of data is very time-consuming. In the next chapter, the adaptation problem is analyzed by performing two exploratory experiments.

4. Problem Analysis

To overcome the language barrier in distributed education, recent works applied speech recognition and machine translation technologies on lectures. The use of simultaneous transcription and translation systems can allow more students and researchers access to exceptional lectures and would help the free distribution of knowledge around the globe. These systems rely on a well-performing automatic speech recognition (ASR). The related work, introduced in chapter 3, has shown that ASR systems can be improved significantly by performing topic adaption. Especially in lectures, the diversity between topics is severe, as shown in the examples in chapter 1. Furthermore, there are hundreds of different lectures at a university. Only for very few of these lectures, the extensive data required for adaptation is already available. However, the effort needed to acquire these data manually is unjustifiable high, due to the many topic-specific data necessary. In this work, it is assumed that for each lecture only the lecture slides are available. The targeted adaptation approach is entirely based on this input data and no further human input is needed during the entire adaptation process.

In the related work, introduced in chapter 3, lecture adaptation was mainly focused on language model adaptation. The vocabulary selection was not described as a challenge. The reason might be that those works were not dealing with multiple diverse lecture topics. Additionally, most of the described approaches used manually selected documents for adaptation. Manual selection of data is reasonable for single lectures. However, due to the extensive costs, it might not be applicable for large-scale adaptation tasks, like adapting a system to every single lecture at a university. For these scenarios, the amount of manual input has to be limited. An approach for vocabulary adaptation used in several related works was adding the vocabulary from the lecture slides to the baseline vocabulary. For this approach, only slides need to be provided by the lecturer. But is this approach sufficient to reach a high vocabulary coverage for many different lecture topics? And what impact does an optimal vocabulary have on speech recognition accuracy? To answer these two questions, two exploratory experiments were performed.

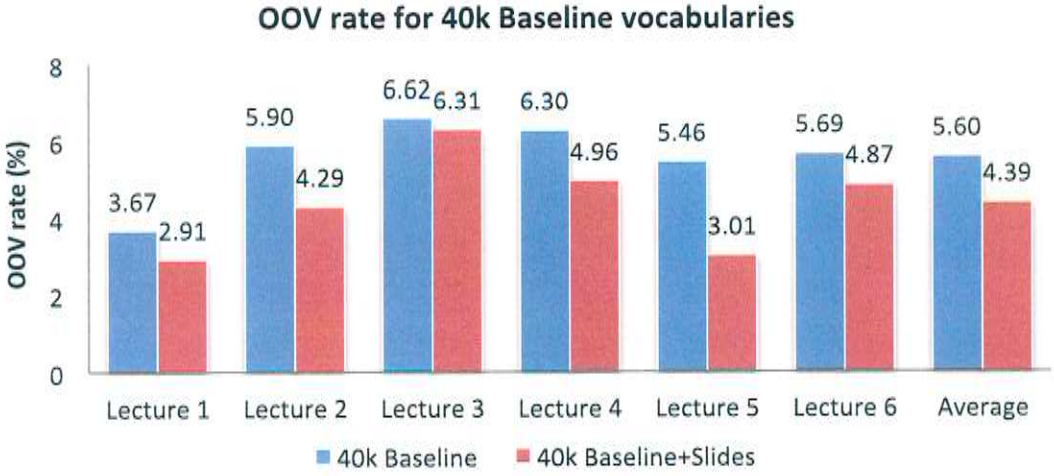

Figure 4.1: Out-of-Vocabulary Rate - 40k Baseline and 40k Baseline+Slides

In this project, the interACT German-English lecture translator, described in chapter 3.1, is available and will be used as our test system. All adaptation approaches will be applied on this system. First, it was tested if including vocabulary from the lecture slides to the recognition vocabulary improves the out-of-vocabulary rate (OOV rate, see section 2.1.1). Three different vocabulary sizes were compared: 40k, 90k, and 300k. The experiments are performed on six German lectures held at Karlsruhe Institute of Technologies. Each lecture has a different topic and a different speaker. The lectures topics were: Data Structures (Lecture 1), Machine Translation (Lecture 2), Mechanics (Lecture 3), Population Geography (Lecture 4), Computer Architecture (Lecture 5), and Copyright Law (Lecture 6). For further information about the data and the setup see chapter 7. The baseline vocabularies with 40k, 90k, and 300k words were selected from combined corpora of broadcast news, parliamentary debates, printed media, and university web data by applying the method described in [StKN10]. When using these vocabularies, the average *out-of-vocabulary rate* (OOV rate) across the six lectures were 5.6% (40k), 3.8% (90k), and 2.8% (300k). Adding vocabulary that occurred in the lecture slides ("Baseline+Slides") relatively reduced OOV rate on average by 22.4%, obtaining average OOV rates of 4.4% (40k), 3.0% (90k), and 2.2% (300k). Figure 4.1 illustrates the OOV rate results for the 40k baseline vocabulary with and without slides added. A detailed breakdown per lecture for all three vocabulary sizes is shown in the appendix in table A.2. This shows that the slides contain words that are uttered during a lecture. However, many words are still missing and the OOV rate after adding the slides' vocabulary was still high.

In the next experiment, the improvement that can be gained by using a perfect vocabulary was determined. Recent work on topic adaptation barely focused on the issue of vocabulary selection, especially for changing topics. Therefore, it seems to be advisable to determine how much enhancement can be gained by improving the vocabulary coverage. In order to do this, the vocabulary was replaced with a vocabulary that includes exactly the words that are uttered during the specific lecture. The speech recognition accuracy of a system that used this "oracle" vocabulary was compared with the a system that used the 40k baseline vocabulary, the same acoustic model, and the same language model data. The results were beyond the expecta-

tions. Just by changing the vocabulary, on average a 10 point lower word error rate (WER) could be achieved, a 30% relative improvement. A detailed breakdown of the results is shown in table 4.1. These results encouraged us to focus on improving vocabulary coverage. The similarity between the six oracle vocabularies was also determined. It was surprising that on average only 5% of the words in the oracle vocabularies were common across the six oracle vocabularies. These 5% common vocabulary represent on average 47% of the words uttered per lecture. Nevertheless, over 50% of the spoken words depended on the lecture topic on the available test set. This shows to which extent the vocabulary can differ between lecture topics and how important an effective adaptation is.

Word Error Rate (%)		
	40k Baseline Vocabulary	Oracle Vocabulary
Lecture 1	43.1	37.8 (12.4%)
Lecture 2	34.9	29.7 (14.7%)
Lecture 3	33.4	21.0 (37.0%)
Lecture 4	28.3	16.6 (41.2%)
Lecture 5	28.4	13.8 (51.3%)
Lecture 6	37.4	28.3 (24.3%)
Average Improvement	-	**30.1%**

Table 4.1: Word Error Rate - Oracle Vocabulary

The goal in this work is to develop an approach that could be used to adapt a speech recognition system to all individual lectures at a university. Therefore, one constraint is that the adaptation needs as little human input as possible. Most related works did not focus on vocabulary selection. However, the exploratory experiments showed that when only a limited amount of data is available per lecture current approaches were not sufficient. An additional challenge is that the vocabularies between different lectures have only a few words in common. Thus, vocabulary selection is a challenge for every new lecture. Nevertheless, the experiment with oracle vocabularies indicated that a significant performance gain is possible. Recent works have adapted speech recognition systems by using lecture slides. One advantage of employing lecture slides is that they are available before the lecture begins and can therefore be used for transcription of live lectures. Another advantage is that the majority of current lectures are held with lecture slides.

For this project, a strong ASR system[1] is given and lecture slides are available for each lecture. The speech recognition vocabulary and the language model are adapted to the lecture topic. This adaptation is entirely based on the unedited lecture slides, no further human input is used. Additional data is automatically collected from the world wide web. The main focus of this adaptation is the selection of a lecture-specific vocabulary by investigating a novel vocabulary ranking approach.

[1]The ASR system of the interACT German-English lecture translator described in chapter 3.1

5. Concept

In this section, the concept of the implemented adaptation approach of the speech recognition component is described. The main focus is hereby on an approach to automatically select a lecture-specific vocabulary. The exploratory experiments have shown significant improvements in speech recognition accuracy just by changing the vocabulary. Although vocabulary selection is a key component for effective adaptation, it has often been overlooked by prior works.

The vocabulary used by a presenter during a lecture can be seen as a combination of two vocabularies as described in the related work in section 3.3: A topic-independent lecture vocabulary, which contains vocabulary common to spontaneous speech, and a topic-specific vocabulary. The proposed approach for vocabulary selection uses a similar breakdown. The lecture-independent vocabulary contains common topic-independent words like stop-words and common words that are used in spontaneous lecture speech. The main goal is to cover words that are used in lectures but not in written text documents. For example, "good morning" and "thank you" are phrases that are commonly uttered during a lecture but they are usually not written in scientific text documents, such as journal articles. The topic-specific vocabulary is selected by using a corpus of topic-specific text documents like in the MIT Spoken Lecture Processing Project introduced in section 3.3. However, in contrast to this related project, not only a few manually selected documents are used but a vast amount of automatically collected documents from the web. This makes the vocabulary selection more challenging but it allows adaptation without further user input.

Figure 5.1 illustrates the proposed approach, which is separated into three steps: Document collection, vocabulary selection, and language model adaptation. Starting with one initial seed document, such as lecture slides, a large corpus of related documents is automatically collected from the world wide web during the document collection step. The vocabulary of the web document corpus is usually too large to be incorporated directly into an ASR system. Therefore, it is necessary to rank the vocabulary to select an optimal vocabulary maintaining the targeted vocabulary size. A novel feature-based vocabulary ranking approach is used to select the active recognition vocabulary. The last step is language model adaptation by using

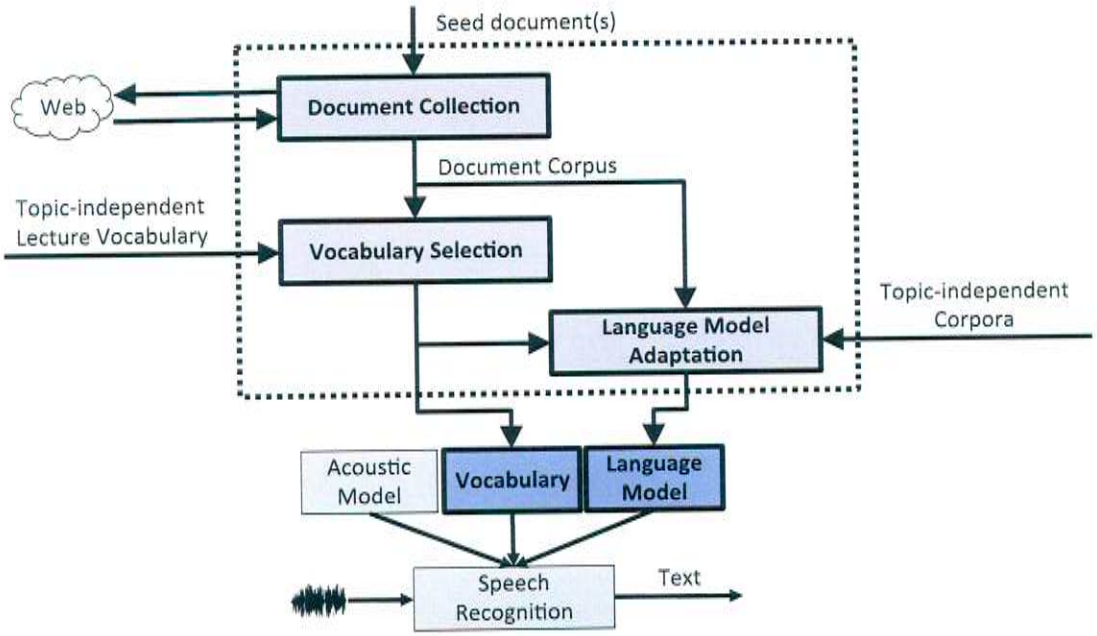

Figure 5.1: Vocabulary Selection and Language Model Adaptation

the new selected vocabulary and the collected document corpus. In the following sections, the three steps, document collection, vocabulary selection, and language model adaptation, are described in detail.

5.1 Document Collection

The document collection process is divided into four steps. It begins with one seed document from which words and key phrases are extracted. Search queries are then automatically generated and a large number of web documents are collected by performing a web-search. Then, language verification is performed on the resulting documents. The single steps of the document collection process are described in detail in the following.

1. **Word Extraction:** The first step in document selection involves extracting text from the seed document. Symbols and punctuation are removed and the text is lowercased and split into individual words. The resulting word-list is then verified against an extremely large dictionary to remove erroneous words that are introduced during the extraction process. In the experimental evaluation (chapter 7), the unigram occurrences from the Google Book n-grams dataset (described in section 2.3) were used.

2. **Query Selection:** Next, search queries are generated from the seed document. Here, short phrases of up to three words that do not contain any topic-independent vocabulary are selected as search queries.

3. **Web-Search:** Web-search[1] is then performed by using this query list. The search is limited to find only results in the source language and for each query,

[1]Search is performed by using the Microsoft Bing search engine.

a fixed number[2] of the highest ranked documents was selected. Then, the text from the resulting documents (web page or PDF file) is extracted.

4. **Language Verification:** For each document, language verification is performed to ensure that it is actually in the required language. When the percentage of topic-independent vocabulary in the document is below 30%, the document is removed from further processing.

5.2 Vocabulary Selection by using Feature-based Ranking

After document collection, the document corpus is used to select a lecture-specific vocabulary. However, if all words in the document corpus are included, the resulting vocabulary is too large to be incorporated directly into an ASR system (in the experimental evaluation vocabularies between 135k and 844k were observed). Thus, a smaller active recognition vocabulary has to be selected. To select words for this smaller vocabulary, a ranking score for each word is computed. Words with the highest score are added to the vocabulary until the desired vocabulary size is reached. Three different *ranking scores* that are based on the different word features (see section 5.3) were applied. For vocabulary ranking, these ranking score functions $s(w)$ were compared to compute the ranking of each word w based on its i^{th} word feature $f_i(w)$:

1. **Single Feature Score:** The score $s_{single,i}(w)$ is based on one single feature $f_i(w)$.

$$s_{single,i}(w) = f_i(w) \tag{5.1}$$

2. **Linear Feature Combination Score:** The score $s_{linear}(w)$ is defined as a linear weighting of two features.

$$s_{linear,i,j}(w) = \alpha \times f_i(w) + (1 - \alpha) \times f_j(w) \tag{5.2}$$

3. **Gaussian Mixture Model Score:** The score $s_{gmm}(w)$ is based on the likelihood ratio of two Gaussian Mixture Models (GMMs). Two GMMs are trained: One for the words that occur in a specific lecture and one for the words that do not occur. The score $s_{gmm}(w)$ is the difference in the log-likelihood of a word feature vector for each of these GMMs. For example with the word feature vector $\mathbf{f}_{i,j}(w) = \begin{pmatrix} f_i(w) & f_j(w) \end{pmatrix}^T$:

$$s_{gmm,i,j}(w) = \log \mathrm{P}_{in}(\mathbf{f}_{i,j}(w)) - \log \mathrm{P}_{out}(\mathbf{f}_{i,j}(w)) \tag{5.3}$$

[2] In the evaluation chapter, section 7.3, the 50 highest ranked documents were used for vocabulary selection and language model adaptation. In a second experiment, a document corpus was collected by including the 500 highest ranked documents. Then, this corpus was used for language model adaptation.

5.3 Features for Vocabulary Selection

The vocabulary ranking scores, described in section 5.2, relies on the features defined in this section. The following annotations for the feature definitions were used:

D	Set of all documents
Q	Set of all queries
W	Set of all words
$d \in D$	Single document
$q \in Q$	Single query
$w \in W$	Single word
D_w	Set of documents that contain the word w
D_q	Set of documents that where found by query q
Q_w	Set of queries that found documents that contained the word w
W_d	Set of all words in the document d
$c_d(w)$	Number of occurrences of the word w in document d

5.3.1 Document Features

For each document, two similarity metrics between the document and the lecture slides are calculated. These similarities are based on the cosine similarity $\text{cosine}(\mathbf{a}, \mathbf{b})$, which is described in section 2.3, eq. 2.11.

A simplified version of the cosine similarity that only compares the words that occur in the slides was used. This modification speeds up the calculation. It also has the effect that if the document contains additional words that are not in the slides the similarity score is not effected. Since the goal is to find documents that contain new previously unseen words, this effect is desired. However, a detailed analysis of this modification might be useful.

1. **Cosine Similarity based on Word Frequency:** Equation 5.4 shows the first similarity metric $\text{WFS}(d)$ between the slides s and the document d. The metric compares the documents based on word frequency vectors, described in section 2.3, eq. 2.10.

$$\text{WFS}(d) = \text{cosine}(\mathbf{freq}_s, \mathbf{freq}_d) \tag{5.4}$$

   where $\mathbf{freq}_s$ is the word frequency vector of the slides s and $\mathbf{freq}_d$ is the word frequency vector of the document d. The word frequency vectors are built based on the words in the slides.

2. **Cosine Similarity based on TF-IDF:** The second similarity metric $\text{TIS}(d)$ (eq. 5.5) is similar to the first, however, instead of the word frequency vectors, it uses approximated TF-IDF vectors (section 2.3, eq. 2.9). IDF is calculated based on the Google Book n-gram dataset (section 2.3).

$$\text{TIS}(d) = \text{cosine}(\mathbf{tfidf}_s, \mathbf{tfidf}_d) \tag{5.5}$$

   where $\mathbf{tfidf}_s$ is the TF-IDF vector of the slides and $\mathbf{tfidf}_d$ is the TF-IDF vector of the document d. Both are built for the vocabulary in the slides.

5.3.2 Query Features

Two features are calculated for the set of resulting documents D_q of each query q. The first feature QWF(q) (eq. 5.6) is the average similarity WFS(d) (eq. 5.4) between the slides and each document $d \in D_q$ found by this query based on the word frequency. The second feature QTI(q) (eq. 5.7) is the average similarity TIS(d) (eq. 5.5) between the slides and each document $d \in D_q$ found by the query based on TF-IDF.

$$\mathrm{QWF}(q) = \frac{\sum_{d \in D_q} \mathrm{WFS}(d)}{|D_q|} \tag{5.6}$$

$$\mathrm{QTI}(q) = \frac{\sum_{d \in D_q} \mathrm{TIS}(d)}{|D_q|} \tag{5.7}$$

5.3.3 Word Features

For each word w, 21 features ($\mathrm{f}_1(w), ..., \mathrm{f}_{21}(w)$) are calculated (equations 5.8 to 5.25). The majority of the features leverages from the document and query features listed above. The main idea of these word features is the higher the value the more relevant is the word for the vocabulary.

1. **DocCount**: Number of documents in which the word occurs.

$$\mathrm{f}_1(w) = |D_w| \tag{5.8}$$

   where D_w is the set of documents in the document corpus that contain the word w.

2. **VocCount**: Number of occurrences in all documents.

$$\mathrm{f}_2(w) = \sum_{d \in D} c_d(w) \tag{5.9}$$

   where D is the set of all documents in the document corpus and $c_d(w)$ is the number of occurrences of the word w in document d.

The next three features (eq. 5.10, 5.11, 5.12) are calculated by using the normalized frequency of the word w in a document.

3. **tfSum**: Sum of term frequencies. This feature is derived Tf-idf. The assumption is that words that occur with a high ratio in the documents of the corpus are more relevant for the vocabulary.

$$\mathrm{f}_3(w) = \sum_{d \in D} \frac{c_d(w)}{\sum_{w_i \in W_d} c_d(w_i)} \tag{5.10}$$

   where W_d is the set of all words in document d.

4. **tfCosineCount**: Sum of term frequencies weighted by the document cosine similarity based on word frequency (eq. 5.4). This feature is similar to tfSum (f_3) with the difference that the ratios are weighted by the similarity of each document to the slides.

$$f_4(w) = \sum_{d \in D} \mathrm{WFS}(d) \frac{c_d(w)}{\sum_{w_i \in W_d} c_d(w_i)} \tag{5.11}$$

5. **tfCosineTfidf**: Sum of term frequencies weighted by the document cosine similarity based on tf-idf (eq. 5.5). Like tfCosineCount (f_4), this feature is a weighted version of tfSum (f_3) but it uses a different similarity measure than tfCosineCount (f_4).

$$f_5(w) = \sum_{d \in D} \mathrm{TIS}(d) \frac{c_d(w)}{\sum_{w_i \in W_d} c_d(w_i)} \tag{5.12}$$

The following three features (eq. 5.13, 5.14, 5.15) are based on the document feature WFS (eq. 5.4) that calculates the cosine similarity between a document and the lecture slides by using word frequency vectors.

6. **DocCosineCount (Max)**: Maximum of the document feature WFS (eq. 5.4) of all documents (D_w) in which the word w occurs. The idea of this feature is that words are more relevant for the vocabulary if they occur in documents that are very similar to the slides. The document similarity is determined by applying the cosine similarity based on word frequency (eq. 5.4).

$$f_6(w) = \max_{d \in D_w}(\mathrm{WFS}(d)) \tag{5.13}$$

7. **DocCosineCount (Min)**: Minimum of the document feature WFS (eq. 5.4) of all documents (D_w) in which the word w occurs. This feature ranks words high that do not occur in documents with low similarity to the slides. Words that do not occur in low ranked documents might be more relevant for the vocabulary.

$$f_7(w) = \min_{d \in D_w}(\mathrm{WFS}(d)) \tag{5.14}$$

8. **DocCosineCount (Avg)**: Average of the document feature WFS (eq. 5.4) of all documents (D_w) in which the word w occurs. If the documents in which the word occurs are on average ranked high, the word might be more likely to occur in the lecture.

$$f_8(w) = \frac{\sum_{d \in D_w} \mathrm{WFS}(d)}{|D_w|} \tag{5.15}$$

The idea of the following three features (eq. 5.16, 5.17, 5.18) is the same as the idea of features $f_6(w)$, $f_7(w)$, and $f_8(w)$. The only difference is that the similarity measure is based on TF-IDF (TIS, eq. 5.5).

9. **DocCosineTfidf (Max)**: Maximum of the document feature TIS (eq. 5.5) of all documents (D_w) in which the word w occurs.

$$f_9(w) = \max_{d \in D_w}(\text{TIS}(d)) \tag{5.16}$$

10. **DocCosineTfidf (Min)**: Minimum of the document feature TIS (eq. 5.5) of all documents (D_w) in which the word w occurs.

$$f_{10}(w) = \min_{d \in D_w}(\text{TIS}(d)) \tag{5.17}$$

11. **DocCosineTfidf (Avg)**: Average of the document feature TIS (eq. 5.5) of all documents (D_w) in which the word w occurs. This feature is similar to feature $f_8(w)$ (DocCosineCount (Avg), eq. 5.15) while using a different similarity measure.

$$f_{11}(w) = \frac{\sum_{d \in D_w} \text{TIS}(d)}{|D_w|} \tag{5.18}$$

The query feature QWF (eq. 5.6) is the foundation of the next three word features (eq. 5.19, 5.20, 5.21). These word features are based on the assumption that if queries have found on average similar documents to the slides, every doc

12. **QueryScoreCount (Max)**: Maximum of query feature QWF (eq. 5.6) of all queries (Q_w) that found the word w.

$$f_{12}(w) = \max_{q \in Q_w}(\text{QWF}(q)) \tag{5.19}$$

where Q_w is the set of all queries that found a document that contains the word w.

13. **QueryScoreCount (Min)**: Minimum of query feature QWF (eq. 5.6) of all queries (Q_w) that found the word w.

$$f_{13}(w) = \min_{q \in Q_w}(\text{QWF}(q)) \tag{5.20}$$

14. **QueryScoreCount (Avg)**: Average of query feature QWF (eq. 5.6) of all queries (Q_w) that found the word w.

$$f_{14}(w) = \frac{\sum_{q \in Q_w} \text{QWF}(q)}{|Q_w|} \tag{5.21}$$

The following 'QueryScoreTfidf'-features (eq. 5.22, 5.23, 5.24) are like the previous three 'QueryScoreCount'-features except for using a different query feature (QTI, eq. 5.7).

15. **QueryScoreTfidf (Max)**: Maximum of query feature QTI (eq. 5.7) of all queries (Q_w) that found the word w.

$$f_{15}(w) = \max_{q \in Q_w}(\text{QTI}(q)) \tag{5.22}$$

16. **QueryScoreTfidf (Min)**: Minimum of query feature QTI (eq. 5.7) of all queries (Q_w) that found the word w.

$$f_{16}(w) = \min_{q \in Q_w}(\mathrm{QTI}(q)) \tag{5.23}$$

17. **QueryScoreTfidf (Avg)**: Average of query feature QTI (eq. 5.7) of all queries (Q_w) that found the word w.

$$f_{17}(w) = \frac{\sum_{q \in Q_w} \mathrm{QTI}(q)}{|Q_w|} \tag{5.24}$$

18. **GoogleBookIDF**: Inverse document frequency (IDF, section 2.3, eq. 2.7) based on the Google Book n-gram dataset.

$$f_{18}(w) = \mathrm{idf}(w) \tag{5.25}$$

The last three features (eq. 5.26, 5.27, 5.28) are the three unigram counts ('match count', 'page count', 'volume count') from the Google Book n-gram dataset (section 2.3).

19. **GoogleBookNgrams (match)**: The word feature f_{19} is the value match_count (total number of occurrences of the word) from the Google Book n-gram dataset (section 2.3) for the word w.

$$f_{19}(w) = \mathrm{match\_count}_{GoogleBook}(w) \tag{5.26}$$

20. **GoogleBookNgrams (page)**: The word feature f_{20} is the value page_count (number of pages that contain the word) from the Google Book n-gram dataset (section 2.3) for the word w.

$$f_{20}(w) = \mathrm{page\_count}_{GoogleBook}(w) \tag{5.27}$$

21. **GoogleBookNgrams (volume)**: The word feature f_{21} is the value volume_count (number of books that contain the word) from the Google Book n-gram dataset (section 2.3) for the word w.

$$f_{21}(w) = \mathrm{volume\_count}_{GoogleBook}(w) \tag{5.28}$$

5.4 Lecture-specific Language Modeling

After the selection of a lecture-specific vocabulary, the lecture-specific document corpus is used for language model adaptation. An approach similar to the web text adaptation approach used by Kawahara et al. (summarized in section 3.3) is used. A topic-independent baseline language model is interpolated with a second lecture-specific language model trained by using the lecture-specific document corpus. This approach is chosen for two reasons. The first reason is that the lecture-specific document corpus does not include sufficient data of spontaneous speech because it

contains mainly written text documents. The second reason is that the lecture-specific document corpus is relatively small and might not contain enough data for a broad n-gram coverage. Both issues can be fixed by including topic-independent corpora that contain various spontaneous speech transcripts and other large text corpora.

The lecture-specific language models are built in three steps. The first step is to train a topic-independent baseline language model based on the n-gram counts in large topic-independent corpus. The resulting language model is called $\mathrm{P}_{(independent\ corpus)}(\mathbf{W})$. The second step is to train a second language model by using the lecture-specific document corpus. This language model is called $\mathrm{P}_{(document\ corpus)}(\mathbf{W})$. Then, these two language models are interpolated to create a lecture-specific language model. The universal interpolation method is described in section 2.1.3. The interpolation is performed by applying a fixed interpolation weight. An interpolation weight of 0.5 was used in the experimental evaluation in chapter 7. This weight has been found to be close to optimum in a small test subset experiment. Formally, the new lecture-specific language model $\mathrm{P}_{adapted}(\mathbf{W})$ is defined as follows:

$$\mathrm{P}_{adapted}(\mathbf{W}) = 0.5 \cdot \mathrm{P}_{(independent\ corpus)}(\mathbf{W}) + 0.5 \cdot \mathrm{P}_{(document\ corpus)}(\mathbf{W}) \qquad (5.29)$$

The all language models were smoothed by using Kneser-Ney smoothing (see section 2.1.3)

5.5 Summary

The proposed adaptation approach starts with a seed document, such as lecture slides. Based on this seed document, similar documents are collected from the world wide web. For vocabulary selection, the words in the collected document corpus are ranked by applying ranking scores best on word features. The highest ranked words are added to a topic-independent vocabulary to form the adapted lecture-specific vocabulary. For language model adaptation, a topic-independent language model is interpolated with a language model based on the collected documents to create the lecture-specific language model. Finally, the lecture-specific vocabulary and language model can be used for lecture-specific speech recognition.

6. Implementation

In this chapter, the implementation of the proposed adaptation approach (chapter 5) is described. Figure 6.1 illustrates the three steps ('document collection', 'vocabulary selection', 'language model adaptation') and shows what was used to implement each part. The document collection was realized by employing a Java program, the vocabulary selection was done by Python scripts, and the language model adaptation was performed by using the Stanford Research Institute language model toolkit (SRILM toolkit, [Stol02]). In the following sections, the implementation is described in more detail.

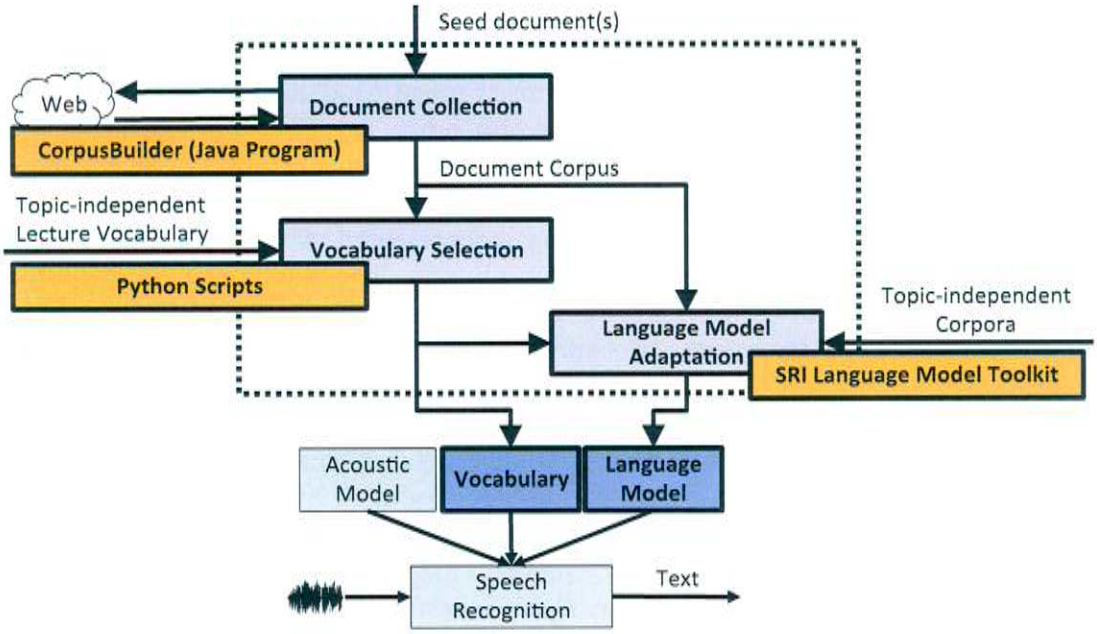

Figure 6.1: Vocabulary Selection and Language Model Adaptation - Implementation

6.1 Document Collection

The document collection was programmed in the programming language Java. The resulting software tool is called *CorpusBuilder*. It handles the collection of a document corpus and the calculation of the features, both described in chapter 5. One topic-independent lecture vocabulary is needed for every language for which this tool should be able to collect a document corpus. As mentioned in chapter 5, this vocabulary is a list of words without topic information, which are commonly used in lecture. For every lecture, the CorpusBuilder needs three input parameters:

1. The location of the slides or a different seed document
2. The desired language of the output document corpus
3. The output directory

When the corpus collection and feature calculation is finished, the CorpusBuilder writes the following output into the output directory:

1. One text file containing all found words with their word features
2. A directory containing one text file for every document in the corpus

Object Structure

The object structure was chosen to make the feature calculation easier and to avoid unnecessary operations. The UML diagram (Unified Modeling Language) of CorpusBuilder can be found at the end of the appendix. A part of the UML diagram is shown in figure 6.2, which is briefly described in this paragraph. A document is represented by the class *Doc*, a word or a short phrase with up to three words is represented by the class *Ngram*, and a query is represented by the class *Query*. Each Doc-object contains 0 to n different Ngram-objects. The class *NgramCount* saves the occurence counts of each word or phrase in a document. A Doc-object has three NgramCount-objects for words, two-word-phrases, and three-word-phrases. Each Ngram-object is found in 0 to n different Doc-objects. Each Query-object finds 0 to n different documents (depending on the number of search results). Every Doc-object is found by 0 to n different Query-objects since seed documents are not found by a search query. Every Ngram-object is uniquely described by its word-string, every Doc-object is uniquely described by its location-string, and every Query is uniquely described by its query-string. Every Query-, Document-, and Ngram-object was handled by the classes *QueryManager*, *DocManager*, and *NgramManager* to ensure that each object is unique. The manager-objects are also used to apply operations on all Query-, Doc-, or Ngram-objects. The class *NgramMap* is just used to sort the words.

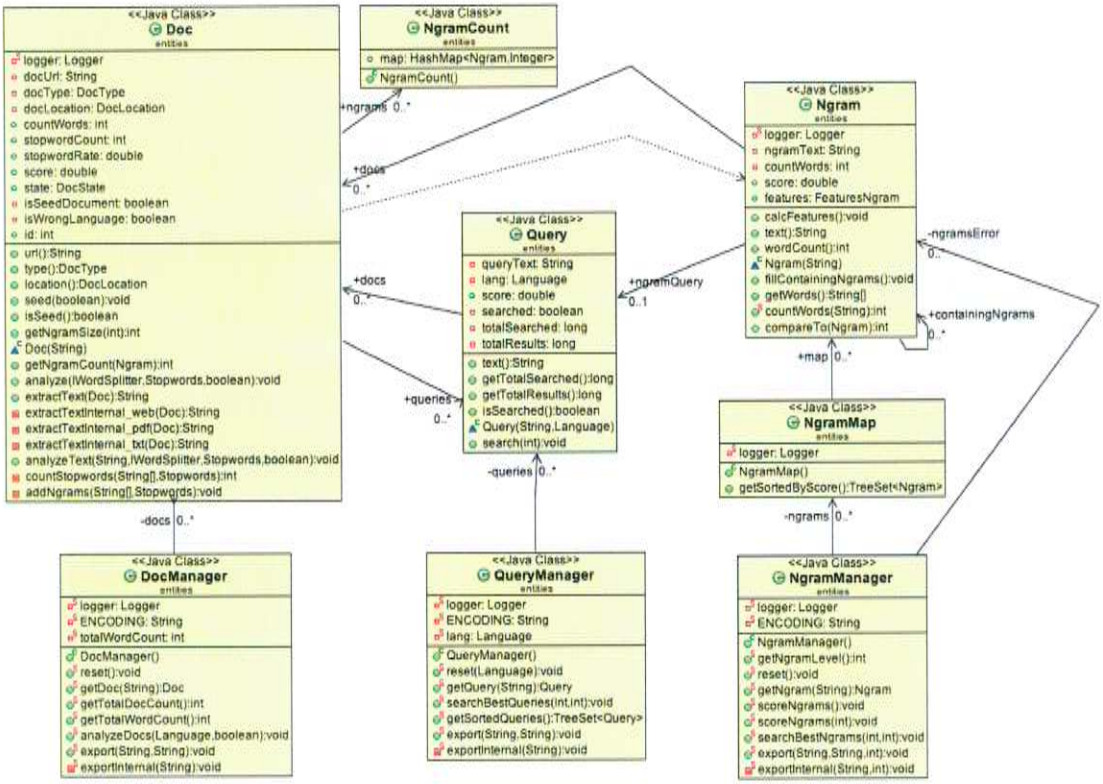

Figure 6.2: Object Structure - Part of UML Diagram

Parallel Processing

It is very time-consuming to perform every step necessary for document collection and feature calculation sequentially. Therefore, parallel processing was performed to speed up the processing. The program starts with extracting text from the seed documents and selecting a list of search queries. It is not necessary to wait for the result of one search query before performing the next search. Thus, these search queries are executed in parallel. The object structure, described in the previous paragraph, makes sure that each search query is only used once. After all search queries are completed, they return a list of multiple URLs that point to the resulting documents. The next step is to download and extract the text from all documents. Again, the processing speed is increased by processing all documents in parallel. Once the text of all documents is available, the words are extracted and the features are calculated. The features of all documents, queries, and words are also calculated in parallel while the object structure ensures that the same calculation is not performed twice. The parallel processing is illustrated in figure 6.3.

6.1.1 Used Packages

An advantage of Java as programming language is that common problems are already solved and the solutions are available as Java libraries. This section gives a short introduction into the packages that were used in the Java program CorpusBuilder to extract the text from PDF documents and webpages, and to perform web search by using the Microsoft Bing search engine.

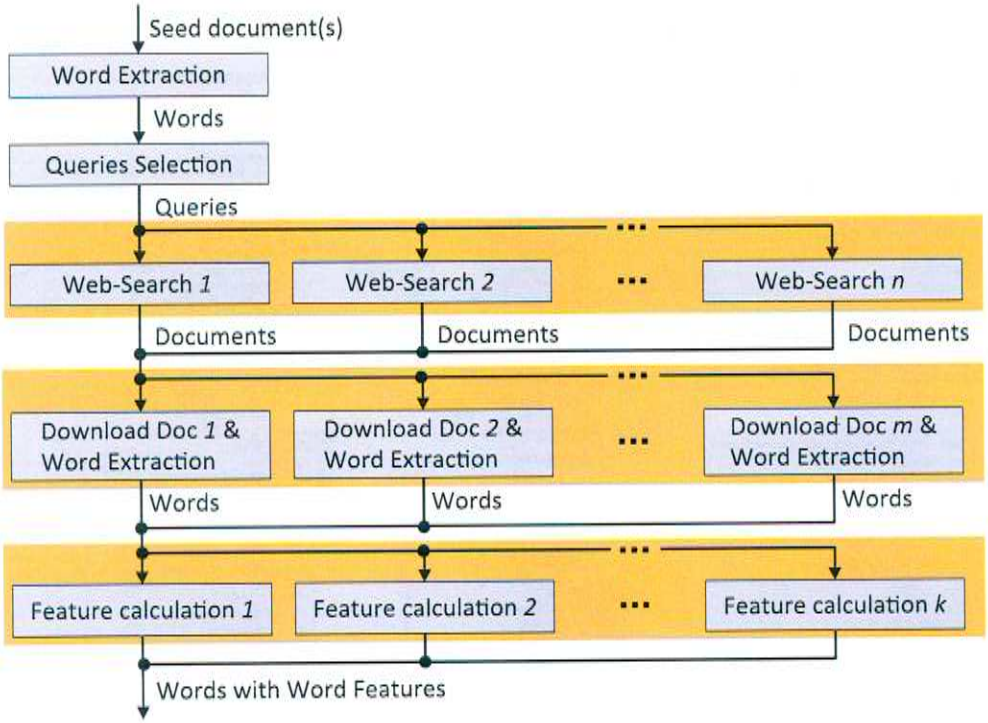

Figure 6.3: Illustration of Parallel Processing in CorpusBuilder

Apache PDFBox™: Java PDF Library

Lecture slides are often available as PDF documents or they can easily be converted into PDF documents. Additionally, many documents on the web are available as PDF documents, such as scientific papers. Therefore, it was important that PDF documents can be processed by the document collection tool CorpusBuilder. The problem is that there are many different versions of PDF documents with different encoding. Luckily, there is already a tool available to handle PDF documents in Java. The Apache PDFBox™ library is an open source Java tool for working with PDF documents, available at http://pdfbox.apache.org. It is constantly updated and offers a variety of different tools to work with PDF documents. For this project, PDFBox was only used to extract the text from PDF documents.

jsoup: Java HTML Parser

Similar to extracting text from PDF documents, extracting text from random web pages is often difficult due to different structures and encodings. Fortunately, there is also a library available to solve this problem. *jsoup*, available at http://jsoup.org/, is a Java library for working with real-world HTML. It provides a very convenient API for extracting and manipulating data. This powerful API was used to extract text from web pages found as search results.

bing-search-java-sdk: A Java wrapper for Bing Search API 2.0

Microsoft Bing was used as search engine for the document collection. Microsoft Bing offers a powerful API, called Bing Search API 2.0, and good search results.

The Bing Search API provides a JSON interface, which accepts search requests in URL format and returns the search results in JSON (JavaScript Object Notation) format. In CorpusBuilder, the Bing Search API was used via the Java wrapper *bing-search-java-sdk*, available at http://code.google.com/p/bing-search-java-sdk/. The bing-search-java-sdk uses this interface. It converts the search query into URL format and sends the request to Bing. All information of the response in JSON format is automatically parsed into an Java object, which makes further processing much easier.

6.2 Vocabulary Selection and Language Model Adaptation

The vocabulary selection is performed by scripts written in Python. Python is a powerful programming language with a clear syntax. Therefore, Python can easily be used for fast development of scripts. The scripts were applied on the output of the CorpusBuilder. They rank the words by using the three ranking scores, described in section 5.2. The Gaussian mixture model ranking was aided by *scikit-learn*, which is a Python module integrating classic machine learning algorithms in Python. It is available at http://scikit-learn.org/.

The language models are trained and interpolated by using the *SRI language modeling toolkit* (SRILM toolkit, [Stol02]) by the Stanford research institute. The n-gram counts are received with the binary *make-batch-counts* and the language model is created based on these counts by using the binary *make-big-lm*. The interpolation of the language models, described in section 5.4, is realized by employing the binary *ngram* with the *-mix-lm* option. For more details about the usages of the SRILM toolkit, see http://www.speech.sri.com/projects/srilm/manpages/.

7. Evaluation

The effectiveness of the proposed method was evaluated on the German speech recognition component of the interACT Simultaneous Lecture Translation system (section 3.1). Figure 7.1 illustrates the adaptation process. The following sections present the evaluation of each step of the adaptation approach. The preparation of the adaptation approach by selecting a topic-independent vocabulary and topic-independent corpora is described in section 7.1. The available lecture data and seed documents are described in section 7.2. The results of the document collections are presented in section 7.3. The different vocabulary selection approaches were evaluated by calculating the out-of-vocabulary rate in section 7.4. The effectiveness of the language model adaptation approach was determined based on the language model perplexity in section 7.5. Finally, the speech recognition performance of the adapted speech recognition system was evaluated in section 7.6.

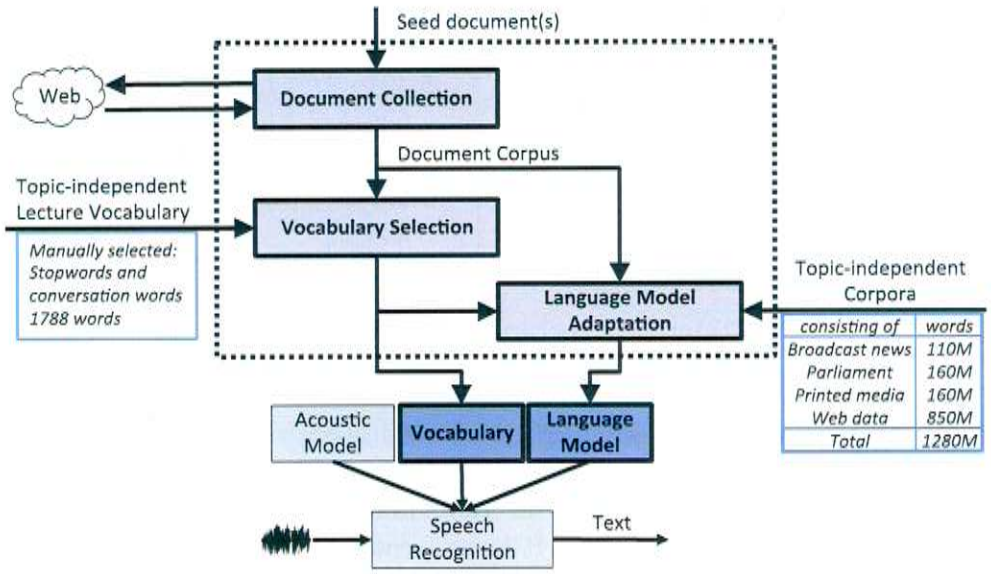

Figure 7.1: Vocabulary Selection and Language Model Adaptation - Evaluation

7.1 Preparation

To evaluate the proposed adaptation approach on German lecture, a topic-independent lecture vocabulary for German lectures and a topic-independent corpora are needed. To select a topic-independent lecture vocabulary, it would be best to analyze many transcribed German lectures of many different topics to determine this vocabulary. Unfortunately, such data were not available for German during the evaluation. Therefore, the topic-independent lecture vocabulary was manually selected from extended stop word lists for German[1,2] and lists of frequent German words[3,4]. The words that were selected had no topic information and included words that are commonly used in spontaneous German speech. The selected German topic-independent lecture vocabulary contained 1788 words.

The language model training corpora of the German lecture translation system (see section 3.1) are used as topic-independent corpora. The topic-independent corpora (1280M words) consisted of broadcast news transcripts (110M words), transcribed parliamentary debates (160M words), printed media (160M words), and web data (850M words).

7.2 Lecture Data

The evaluation was performed on six lectures held at Karlsruhe Institute of Technology, in 2009 and 2010. The lectures consisted of a variety of topics: Data Structures (Lecture 1), Machine Translation (Lecture 2), Mechanics (Lecture 3), Population Geography (Lecture 4), Computer Architecture (Lecture 5), and Copyright Law (Lecture 6). Each of the six lectures was held by a different speaker in German. The lectures 1, 2, and 4 were completely transcribed while lectures 3, 5, and 6 were only partially transcribed. The evaluation is performed on a total of 5.7 hours of transcribed lecture audio. For each lecture, the topic, the length of the transcribed audio, the number of spoken words, and the used language are shown in table 7.1.

	Topic	Duration	#words	Language
Lecture 1	Data Structures	5,491 s	11,495	German
Lecture 2	Machine Translation	5,002 s	11,959	German
Lecture 3	Mechanics	766 s	1,315	German
Lecture 4	Population Geography	5,253 s	12,168	German
Lecture 5	Computer Architecture	296 s	531	German
Lecture 6	Copyright Law	3,710 s	9,162	German

Table 7.1: Lecture Data - Recording

For each lecture, the lecturer provided the set of lecture slides that he or she used during the lecture. Table 7.2 shows for each lecture the number of slides provided, the total number of words in those slides, the number of unique words in the slides, and the language used in the slides. It is noticeable that the lecturer of lecture 2

[1] http://www.ranks.nl/stopwords/german.html
[2] http://solariz.de/649/deutsche-stopwords.htm
[3] http://german.about.com/library/blwfreq01.htm
[4] http://wortschatz.uni-leipzig.de/html/wliste.html

used English slides even though she presented in German. Those slides were used to collect a German document corpus in the same manner in which German slides were used.

	#slides	#words	#words (unique)	Language
Lecture 1	109	5,866	875	German
Lecture 2	42	2,233	449	English
Lecture 3	3	165	72	German
Lecture 4	61	1,326	469	German
Lecture 5	89	4,781	1,222	German
Lecture 6	16	1,134	520	German

Table 7.2: Lecture Data - Lecture Slides

7.3 Document Collection

The slides provided for each lecture were used as seed document to collect related documents in German by applying the document collection approach described in section 5.1. The provided slides were not altered in any way before extracting the text for document collection. The resulting document corpora of the six lectures varied in size. This is due to the different number of words in the slides and therefore different number of search queries. Furthermore, some search queries did not receive the maximum amount of 50 search results because the total number of results was lower. Table 7.3 shows for each document corpus the number of documents, the number of words, and the size of the corpus in megabytes (MBytes). Additionally, an extended document collection searching for up to 500 documents per query was performed. These extended document corpora were collected while working on the language model adaptation to have more data. Thus, the extended corpora were not used for vocabulary selection. To get 500 documents per query, it was necessary to sent 10 separate requests to Bing because the Bing Search API only returns up to 50 documents per search requests. This slowed down the extended document collection significantly. The extended document corpora of the six lectures varied in the same manner as the smaller document corpora. A detailed breakdown of extended document corpora is shown in the appendix in table A.1.

	#documents	#words	#words (unique)	size (MBytes)
Lecture 1	21,254	38,889,465	631,217	253
Lecture 2	5,259	8,418,297	212,614	39
Lecture 3	1,469	4,695,643	134,939	29
Lecture 4	10,430	21,023,728	458,238	140
Lecture 5	33,654	64,831,722	844,369	419
Lecture 6	14,314	30,837,822	688,894	206

Table 7.3: Document Collection Results (Up to 50 Results per Query)

7.4 Vocabulary Selection

After the document collection, the proposed vocabulary selection approach was evaluated by calculating the out-of-vocabulary (OOV) rate for different vocabularies. The results of the baseline vocabulary on each of the six lectures was shown in chapter 4, figure 4.1. A detailed breakdown per lecture is shown in the appendix in table A.2.

7.4.1 Boundaries

The OOV rate of the selected lecture-specific vocabularies is limited by two boundaries, the OOV rate of the topic-independent lecture vocabulary and the OOV rate of the Google Book unigrams. Every selected vocabulary contains the topic-independent lecture vocabulary. Thus, the highest OOV rate a vocabulary can achieve is the OOV rate of the topic-independent vocabulary. On the six lecture the average OOV rate of the topic-independent vocabulary was 17.59%. Additionally, the new vocabularies were verified against an extremely large vocabulary to remove erroneous words that are introduced during the extraction process. In this evaluation, the unigram occurrences from the Google Book n-gram dataset were utilized (described in section 2.3), which in total contains about 3 million unique word entries. The Google Book unigram vocabulary was tested to determine the lowest OOV rate that could be achieved. The average OOV rate of the Google Book unigrams on the six lectures was 0.51%. Detailed results for this two vocabularies are shown in table 7.4.

	Out-of-Vocabulary Rate (%)	
	Topic-Independent Lecture Vocabulary (1788 words)	Google Book unigrams (≈3M words)
Lecture 1	12.90	0.43
Lecture 2	15.35	1.20
Lecture 3	16.50	0.23
Lecture 4	17.75	0.22
Lecture 5	22.41	0.19
Lecture 6	20.62	0.80
Average	17.59	0.51

Table 7.4: Out-of-Vocabulary Rate - Topic-Independent Lecture Vocabulary and Google Book Unigrams

7.4.2 Feature-based Vocabulary Selection

The usefulness of the three different vocabulary ranking scores and the 21 different word features was evaluated by calculating the OOV rate for vocabularies selected by the different methods. This evaluation was performed only on lectures 1-4 as transcripts of lectures 5 and 6 were not available when this evaluation took place. The document corpora that were collected by searching for up to 50 documents per search query were used for vocabulary selection.

Single Feature Score Ranking

In the first experiments, vocabularies were selected by applying the single feature ranking score (see section 5.2, eq. 5.1). The average OOV rate of 40k vocabularies that were selected by using the single feature scores of all 21 features is shown in figure 7.2. The lowest OOV rate was obtained by employing feature 1, DocCount (f_1), delivering average OOV rates of 2.4% (40k), 1.6% (90k), and 1.1% (300k). The feature 2, VocCount (f_2), obtained similar OOV rates, on average 2.6% (40k), 1.7% (90k), and 1.1% (300k). Vocabulary selection based on either of these two features leads to a significantly lower OOV rate than the OOV rate of the three baseline systems. The single feature score ranking with the DocCount feature improved the baseline OOV rate on average by 56.8% while maintaining the same vocabulary size. A detailed breakdown of all single feature results for all features, vocabulary sizes, and lectures is shown in the appendix, table A.3. The vocabularies selected by using feature 1, DocCount, or feature 2, VocCount, had a better vocabulary coverage than the baseline vocabulary with the same size. For lecture 1 and 3, the vocabulary coverage was higher by using the 40k DocCount vocabulary than by using the 300k baseline vocabulary. The DocCount vocabularies for lectures 1, 3, and 4 had also a better vocabulary coverage than the baseline vocabularies with slides added. For lecture 2, this was only true for the 40k case while the 90k and 300k baseline+slides vocabularies obtained a lower OOV rate. For lecture 2 and 3, the OOV rate of all 300k single feature score vocabularies was the same because the vocabulary size for the document corpora were below 300k (see section 7.3, table 7.3).

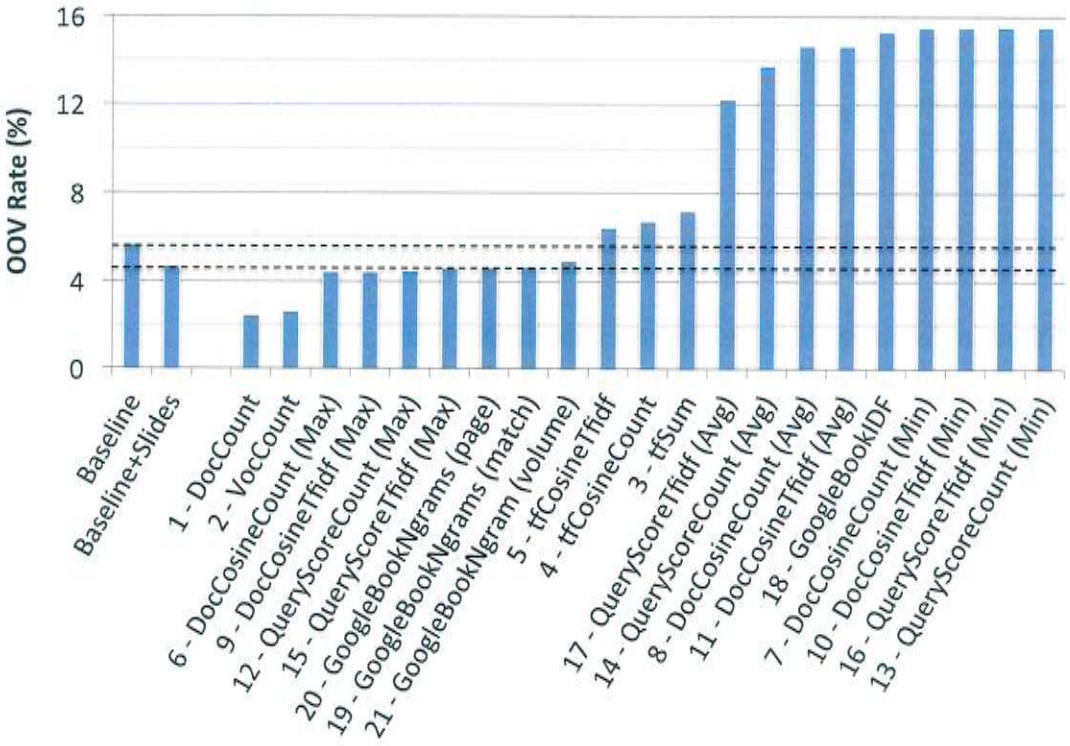

Figure 7.2: Average OOV Rate for all Features - 40k Vocabulary

Linear Feature Combination Score Ranking

The effectiveness of combining multiple features for vocabulary ranking was investigated in the next experiments. Pairs of features were linearly combined by applying the linear feature combination score (section 5.2, eq. 5.2) with $\alpha \in \{0.1, 0.2 \ldots, 0.9\}$. All 210 feature pairs of the 21 features were evaluated, leading to 1890 feature combinations. For most feature combinations, the linear combination vocabularies obtained an OOV rate that was between the OOV rates of the two single feature vocabularies. However, combining DocCount and VocCount with $\alpha = 0.5$ ("Doc+VocCount") obtained in most cases a better or equal vocabulary coverage and an average reduction of OOV rate of 1% compared to using the DocCount feature alone, obtaining average OOV rates of 2.32% (40k), 1.62% (90k), and 1.14% (300k). The largest relative reduction in OOV rate was 84.9%, which was obtained on lecture 3 for a 300k vocabulary, reducing the OOV rate from 5.0% (Baseline) to 0.76% (Doc+VocCount). The results of DocCount, VocCount, and Doc+VocCount, are compared in table 7.5. Overall, the result of the linear score Doc+VocCount is better.

Out-of-Vocabulary Rate (%)					
40k	Lecture 1	Lecture 2	Lecture 3	Lecture 4	Average
DocCount	**1.50**	3.66	1.52	**2.86**	2.39
VocCount	1.72	3.57	1.67	3.31	2.57
Doc+VocCount	**1.50**	**3.55**	**1.37**	**2.86**	**2.32**
90k	Lecture 1	Lecture 2	Lecture 3	Lecture 4	Average
DocCount	**1.10**	2.83	**0.84**	1.73	1.63
VocCount	1.22	**2.80**	**0.84**	1.98	1.71
Doc+VocCount	**1.10**	**2.80**	**0.84**	**1.72**	**1.62**
300k	Lecture 1	Lecture 2	Lecture 3	Lecture 4	Average
DocCount	0.73	**2.12**	**0.76**	**0.92**	**1.13**
VocCount	**0.70**	**2.12**	**0.76**	0.98	1.14
Doc+VocCount	0.75	**2.12**	**0.76**	0.94	1.14

Table 7.5: Out-of-Vocabulary Rate - DocCount, VocCount, and Doc+VocCount

Gaussian Mixture Model Score Ranking

Next, the GMM-based vocabulary ranking (Gaussian Mixture Model Score, section 5.2, eq. 5.3) was evaluated. This evaluation began with an oracle test, by training the GMMs based on a lecture transcript and testing it on the same lecture. The evaluation was started with an oracle test to determine the best parameters and to ensure that this approach could lead to better results. Two GMMs were trained, one GMM for words that did occur in the transcript and one GMM for words that did not occur. All feature-pairs of the 21 features on all four lectures were evaluated. GMMs with one, two, three, and four components were tested. The goal of this oracle test was to identify a feature combination that received a lower OOV rate than the Doc+VocCount ranking for all four lectures. Although slight improvements were gained for specific lectures, no feature-pair consistently improved performance across all lectures. Since it was not possible to identify a promising feature-pair for Gaussian mixture model score ranking, this approach was not investigated any further.

Selected Ranking Score

After comparing the different ranking scores and features, it was clear that the linear score Doc+VocCount achieved the best results. Therefore, vocabularies selected by using this score were used in the following evaluations. Fig. 7.3 shows the effectiveness of the proposed linear score Doc+VocCount on the six lectures compared to the baseline over varying vocabulary sizes. The proposed approach relatively reduced the OOV rate on average by 50.6%, 49.3%, and 59.2% for the 40k, 90k, and 300k systems per lecture. More significantly the 40k vocabulary selected with the proposed approach obtained on average an OOV rate similar to the average OOV rate of the 300k baseline system, showing the effectiveness of this approach. A detailed breakdown per lecture is shown in the appendix in table A.4.

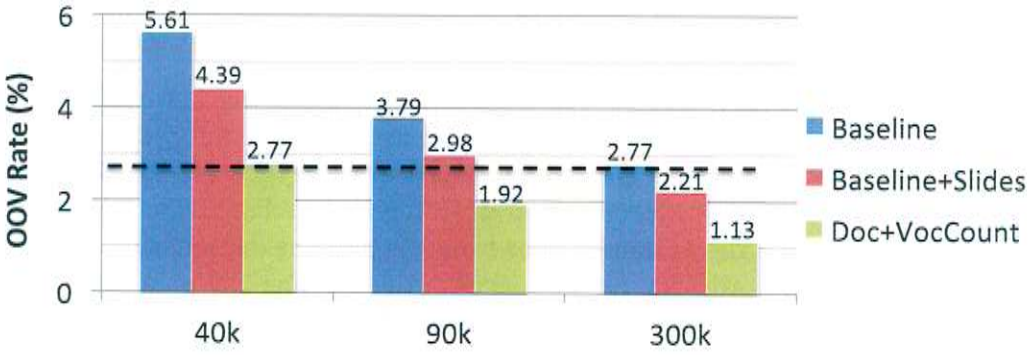

Figure 7.3: Average OOV Rate of Baseline compared with Doc+VocCount

7.5 Lecture-dependent Language Model Adaptation

After the vocabulary selection, the method described in section 5.4 was used to train a lecture-specific language model (LM) by employing the topic-independent corpora (see section 7.1) and one of the lecture-specific corpora collected in section 7.3. For each lecture, the lecture-specific vocabulary selected in the last section were used. The SRILM [Stol02] toolkit was used for LM training and LM interpolation. Two different lecture-specific LMs were built: One based on the lecture-specific document corpus collected by searching for up to 50 document per query ("search50"), and one based on the lecture-specific document corpus collected by searching for up to 500 documents ("search500"). These lecture-specific LMs are compared with the baseline LMs in terms of perplexity on the specific lecture transcripts. The results for the 40k vocabularies are shown in tables 7.6, the results for the 90k and 300k vocabularies are shown in the appendix in table A.5, and in table A.6. The lecture-specific LMs obtained a significantly lower perplexity compared to the baseline lecture-independent model. However, the larger document corpus "search500" did not improve language model perplexity. On average for the 40k vocabularies, the relative improvement was 23.2% with the "Search50" corpus and 22.8% with the "Search500" corpus.

Language Model Perplexity			
	Baseline	Adapt LM	Adapt LM
Documents per query	-	50 ("Search50")	500 ("Search500")
Lecture 1	344.0	261.4 (24.0%)	266.2 (22.6%)
Lecture 2	352.0	285.7 (18.8%)	291.3 (17.3%)
Lecture 3	325.0	199.9 (38.5%)	192.2 (40.9%)
Lecture 4	247.1	210.0 (15.0%)	207.1 (16.2%)
Lecture 5	274.3	170.0 (38.0%)	184.0 (32.9%)
Lecture 6	241.3	229.9 (4.7%)	225.1 (6.7%)
Avg. Improvement	-	**23.2%**	**22.8%**

Table 7.6: Language Model Perplexity - 40k Vocabulary

7.6 Lecture-dependent Speech Recognition

Speech recognition of each lecture was performed by using the automatic speech recognition (ASR) component of the interACT lecture translation system (section 3.1). The ASR component used the *Janus Recognition Toolkit* (JRTk) [SMFW01]. The acoustic model training was described in [KWKN+08]. For the lecture-specific recognition dictionaries, the pronunciations from the baseline dictionaries were used. The pronunciations of words that did not occur in the baseline dictionaries were generated automatically by employing the Festival Speech Synthesis System [TaBC98].

The speech recognition accuracy of four different systems was evaluated, the unadapted topic-independent baseline system ("Baseline"), a system with adapted lecture-specific vocabulary and baseline language model ("Only Vocab"), a system with adapted lecture-specific language model and baseline vocabulary ("Only LM"), and the proposed system with adapted lecture-specific vocabulary and language model ("Both Vocab & LM"). Unsupervised speaker adaptation was performed for each lecture. The WER results of the four systems with a 40k vocabulary and language model adaptation based on the "search50"-document corpora are shown in table 7.7. The lecture-independent baseline system obtained an WER of 35.6% for all the six lectures combined. The proposed approach to adapt the vocabulary and the language model lowered the WER of all six lectures. The combined WER was 32.8% (a 7.9% relative reduction compared with the baseline). Applying only vocabulary adaptation or only language model adaptation showed that the adaptation of the vocabulary had the greater impact. When only vocabulary selection (described in section 5.2) based on the linear feature combination score (Doc+VocCount) was performed, a WER of 34.1% was obtained for all six lectures combined, a relative reduction of 4.2% compared to the baseline system. A WER of 35.4% for all lecture combined (a relative improvement of 0.4%) was received when only applying language model adaptation (described in section 5.4). The lecture-specific vocabularies led to bigger improvement in WER than the language model adaptation. But, the biggest gain was obtained by combining both, vocabulary selection and language model adaptation. On average, the WER was lowered by 12.5% per lecture by using the proposed combination of vocabulary and language model adaptation. The proposed approach improved speech recognition accuracy on all six lectures even though the improvement varied between the lectures. The WER results of systems with larger vocabularies, and language model adaptation based on the large docu-

ment corpora showed similar results with similar fluctuations. A detailed breakdown can be found in the appendix in tables A.14, A.15, A.16, A.17, A.18. The result of tests with speaker unadapted acoustic models are shown in the appendix in tables A.7, A.8, A.9, A.10, A.11, A.12.

Word Error Rate (%)							
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	43.1	42.4	(1.7%)	42.7	(1.0%)	41.0	(5.0%)
Lecture 2	34.9	35.7	(-2.3%)	34.3	(1.6%)	33.9	(2.7%)
Lecture 3	33.4	27.3	(18.2%)	34.7	(-3.8%)	27.5	(17.6%)
Lecture 4	28.3	23.9	(15.6%)	28.5	(-0.6%)	22.7	(20.0%)
Lecture 5	28.4	28.8	(-1.3%)	25.5	(10.4%)	21.2	(25.3%)
Lecture 6	37.4	36.4	(2.6%)	37.6	(-0.6%)	35.7	(4.4%)
Lectures 1-6	35.6	34.1	(4.2%)	35.4	(0.4%)	32.8	(7.9%)
Avg. improve. per lecture	-	**5.8%**		**1.3%**		**12.5%**	

Table 7.7: Word Error Rate - Speaker Adaptation - 40k Vocabulary - "Search50"-Language Model

7.7 Discussion

Although improvements in WER were achieved across all test lectures, the improvement of some lectures was relatively lower. For example on lecture 2, only a relative improvement of 2.7% in WER was obtained compared to the baseline. A possible explanation for this small improvement can be the high percentage of English words in the German lecture (see table 7.8). In the available test lectures, lecture 2 had the highest percentage of English words (7.42% of all spoken words). In some German lectures, professors use many technical terms in English. It was shown by Kolss et al. in [KWKN+08] that recognition of such words are often a problem due to different phonemes and pronunciation rules in the English language. Another issue is the combination of English words and German declension or conjugation, which often occurs in today's spoken German. An example is the word "boost**en**", which was used in lecture 2. It is a combination of the English verb "to boost" and the German ending of an infinitive "-en". Another example is the word "down**ge**load**et**", which is a combination of the English verb "to download" and German conjugation rules. These two words are used in today's spoken German but they are usually not found in written German documents. Therefore, these examples indicate a new problem that occurs in speech recognition of German lectures. Solving these problems might improve the speech recognition accuracy of lectures, such as lecture 2.

Percentage of English words in German transcripts					
Lecture 1	Lecture 2	Lecture 3	Lecture 4	Lecture 5	Lecture 6
2.49%	7.42%	0.15%	0.42%	1.13%	0.82%

Table 7.8: Percentage of English Words in the six German Test Lectures.

7.8 Summary

The proposed approach was successfully evaluated on six German lectures with different lecture topics and different speakers. By applying the vocabulary selection approach described in section 5.2, the out-of-vocabulary rate was relatively improved on average by 53.0% per lecture and vocabulary size compared to a lecture-independent baseline vocabulary. The language model adaptation approach described in section 5.4 lowered the language model perplexity on average by 23.2% per lecture compared to a lecture-independent baseline. By combining the vocabulary selection and language model adaptation, the word error rate was reduced on average by 12.5% per lecture compared to a lecture-independent baseline system by using a 40k vocabulary .

8. Conclusion

Speech recognition technologies and especially the introduction of the interACT lecture translation system have shown that speech recognition of lectures can reduce language barriers in today's education. However, due to the variety of different lecture topics and the huge differences between each topic, current systems perform poorly. To solve this problem, related work has shown significant improvements in speech recognition accuracy by adapting the speech recognition system to the topic of the lecture. Nevertheless, the existing approaches are either too time-consuming or ineffective. The approach proposed in this work is different. Assuming that one document, such as lecture slides, is available for every lecture, the proposed approach automatically adapts a speech recognition system to the topic of the current lecture without any further human input. Based on the initial document, related documents are collected from the world wide web. The adaptation is focused on the selection of a lecture-specific vocabulary from the collected documents. The vocabulary is selected by using a novel unsupervised vocabulary selection approach that uses feature-based ranking scores. Additionally, the language model is adapted to the lecture topic by leveraging the collected documents. During the evaluation of the proposed approach on six German lectures with different topics and speakers, the effectiveness of this adaptation approach was shown in comparison to a lecture-independent baseline. By selecting a lecture-specific vocabulary for each lecture, the out-of-vocabulary rate was relatively improved on average by 53.0% per lecture compared to a lecture-independent baseline vocabulary. Furthermore, the language model perplexity was lowered on average by 23.0% per lecture by using the proposed adaptation method. Finally, the word error rate was reduced by 12.5% per lecture with the help of a lecture-specific vocabulary and language model for each lecture compared to a lecture-independent baseline system.

The proposed adaptation method has shown improvement for every lecture in the test set. However, the improvement varied considerably between each lecture. In future works, this variation should be examined further on large test sets. Additionally, the proposed method relies on a vocabulary verification to remove erroneous words by using the large Google Book n-gram data set. Although this data set contains a huge number of words, it did not contain all words that were spoken in the test lectures. The reason for that might be a mismatch between written and spoken words. For future works, it could be useful to investigate the use of different verification data sets.

A. Appendix

	#documents	#words	#words (unique)	Size (MBytes)
Lecture 1	92,331	214,863,487	1,441,029	1,340
Lecture 2	12,754	34,421,841	429,582	148
Lecture 3	5,725	65,069,621	411,433	289
Lecture 4	44,365	102,523,661	1,094,376	707
Lecture 5	190,852	366,410,550	1,613,089	2,342
Lecture 6	83,551	170,852,776	1,314,576	1,132

Table A.1: Extended Document Collection Results (up to 500 Results per Query)

Vocabulary Selection

Baseline

Out-of-Vocabulary Rate (%)						
Baseline Size	40k	40k	90k	90k	300k	300k
Slides added	-	Yes	-	Yes	-	Yes
Lecture 1	3.67	2.91	2.02	1.71	1.70	1.43
Lecture 2	5.90	4.29	3.80	2.36	3.12	1.97
Lecture 3	6.62	6.31	5.48	5.17	5.02	4.71
Lecture 4	6.30	4.96	4.79	3.64	2.24	1.83
Lecture 5	5.46	3.01	2.45	1.51	2.07	1.32
Lecture 6	5.69	4.87	4.20	3.46	2.47	2.00
Average	5.60	4.39	3.79	2.98	2.77	2.21

Table A.2: Out-of-Vocabulary Rate - Baseline vocabularies

Feature-based Vocabulary Selection

Out-of-Vocabulary Rate (%)												
	Lecture 1			Lecture 2			Lecture 3			Lecture 4		
	40k	90k	300k	40k	90k	300k	40k	90k	300k	40k	90k	300k
Baseline	3.67	2.02	1.70	5.90	3.80	3.12	6.62	5.48	5.02	6.30	4.79	2.24
Baseline+Slides	2.91	1.71	1.43	4.29	2.36	1.97	6.31	5.17	4.71	4.96	3.64	1.83
1 - DocCount	1.50	1.10	0.73	3.66	2.83	2.12	1.52	0.84	0.76	2.86	1.73	0.92
2 - VocCount	1.72	1.22	0.70	3.57	2.80	2.12	1.67	0.84	0.76	3.31	1.98	0.98
6 - DocCosineCount (Max)	2.29	1.50	0.78	5.36	3.38	2.12	4.34	1.52	0.76	5.60	2.85	0.98
9 - DocCosineTfidf (Max)	2.29	1.50	0.78	5.36	3.38	2.12	4.34	1.52	0.76	5.60	2.85	0.98
12 - QueryScoreCount (Max)	3.01	1.48	0.84	6.77	4.37	2.12	2.97	1.45	0.76	5.02	2.82	1.04
15 - QueryScoreTfidf (Max)	3.27	1.39	0.80	6.66	4.05	2.12	2.97	1.14	0.76	5.38	2.60	1.04
20 - GoogleBookNgrams (page)	3.41	2.51	1.11	4.69	3.63	2.12	5.86	2.28	0.76	4.44	2.40	1.02
19 - GoogleBookNgrams (match)	3.42	2.51	1.05	4.70	3.40	2.12	5.78	2.28	0.76	4.57	2.43	1.03
21 - GoogleBookNgram (volume)	3.97	2.68	1.17	4.96	3.55	2.12	5.94	2.44	0.76	4.78	2.51	1.04
5 - tfCosineTfidf	4.92	3.26	0.97	5.97	3.46	2.12	2.66	0.91	0.76	12.18	6.93	1.00
4 - tfCosineCount	5.08	3.44	0.99	6.40	3.63	2.12	2.89	0.91	0.76	12.52	7.14	1.02
3 - tfSum	5.94	4.37	0.90	6.09	3.42	2.12	3.12	0.91	0.76	13.58	8.82	1.00
17 - QueryScoreTfidf (Avg)	10.11	10.11	4.43	13.57	8.26	2.12	8.30	2.05	0.76	16.90	15.31	1.51
14 - QueryScoreCount (Avg)	11.14	11.14	5.12	13.81	8.58	2.12	12.40	2.97	0.76	17.62	16.84	1.60
8 - DocCosineCount (Avg)	11.74	10.15	3.98	13.25	7.84	2.12	16.06	3.20	0.76	17.49	16.75	1.25
11 - DocCosineTfidf (Avg)	11.74	10.15	3.98	13.25	7.84	2.12	16.06	3.20	0.76	17.49	16.75	1.25
18 - GoogleBookIDF	12.90	12.89	12.46	15.19	14.68	2.12	15.37	11.64	0.76	17.74	17.60	16.70
7 - DocCosineCount (Min)	12.89	12.88	12.66	15.10	14.56	2.12	16.29	15.07	0.76	17.67	17.59	16.17
10 - DocCosineTfidf (Min)	12.89	12.88	12.66	15.10	14.56	2.12	16.29	15.07	0.76	17.67	17.59	16.17
16 - QueryScoreTfidf (Min)	12.88	12.88	12.50	15.04	14.64	2.12	16.44	14.31	0.76	17.68	17.55	16.17
13 - QueryScoreCount (Min)	12.88	12.88	12.50	15.13	14.70	2.12	16.44	14.46	0.76	17.67	17.60	16.20

Table A.3: Out-of-Vocabulary Rate - Single Feature Score Ranking

Out-of-Vocabulary Rate (%)								
		Lecture						
		1	2	3	4	5	6	Average
Baseline	40k	3.67	5.90	6.62	6.30	5.46	5.69	5.60
Baseline+Slides	40k	2.91	4.29	6.31	4.96	3.01	4.87	4.39
Doc+VocCount	40k	1.50	3.55	1.37	2.86	3.39	3.96	2.77
Baseline	90k	2.02	3.80	5.48	4.79	2.45	4.20	3.79
Baseline+Slides	90k	1.71	2.36	5.17	3.64	1.51	3.46	2.98
Doc+VocCount	90k	1.10	2.80	0.84	1.72	2.26	2.82	1.92
Baseline	300k	1.70	3.12	5.02	2.24	2.07	2.47	2.77
Baseline+Slides	300k	1.43	1.97	4.71	1.83	1.32	2.00	2.21
Doc+VocCount	300k	0.75	2.12	0.76	0.94	0.56	1.62	1.12

Table A.4: Out-of-Vocabulary Rate - Baseline, Baseline+Slides, and Doc+VocCount

Lecture-dependent Language Model Adaptation

Language Model Perplexity			
	Baseline	Adapt LM	Adapt LM
Documents per query	-	50 ("Search50")	500 ("Search500")
Lecture 1	356.1	270.0 (24.2%)	275.6 (22.6%)
Lecture 2	382.1	305.7 (20.0%)	312.6 (18.2%)
Lecture 3	342.6	207.3 (39.5%)	200.3 (41.6%)
Lecture 4	277.5	232.3 (16.3%)	229.5 (17.3%)
Lecture 5	305.3	188.6 (38.2%)	204.5 (33.0%)
Lecture 6	269.3	256.9 (4.6%)	251.1 (6.8%)
Avg. Improvement	-	**23.8%**	**23.2%**

Table A.5: Language Model Perplexity - 90k Vocabulary

Language Model Perplexity			
	Baseline	Adapt LM	Adapt LM
Documents per query	-	50 ("Search50")	500 ("Search500")
Lecture 1	369.2	279.0 (24.4%)	285.1 (22.8%)
Lecture 2	409.2	325.9 (20.4%)	333.3 (18.6%)
Lecture 3	344.5	208.2 (39.6%)	201.4 (41.5%)
Lecture 4	302.9	251.1 (17.1%)	248.1 (18.1%)
Lecture 5	367.9	222.3 (39.6%)	237.8 (35.4%)
Lecture 6	302.1	287.5 (4.8%)	281.0 (7.0%)
Avg. Improvement	-	**24.3%**	**23.9%**

Table A.6: Language Model Perplexity - 300k Vocabulary

Lecture-dependent Speech Recognition

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	51.2	50.7	(1.0%)	51.4	(-0.4%)	49.6	(3.1%)
Lecture 2	39.0	39.4	(-1.1%)	38.3	(1.6%)	38.0	(2.5%)
Lecture 3	38.3	33.4	(12.8%)	37.6	(1.9%)	30.9	(19.4%)
Lecture 4	31.3	26.8	(14.4%)	31.4	(-0.5%)	25.7	(18.0%)
Lecture 5	31.6	30.6	(2.9%)	29.2	(7.6%)	26.0	(17.6%)
Lecture 6	46.1	45.1	(2.0%)	45.8	(0.5%)	44.4	(3.5%)
Lectures 1-6	41.3	39.7	(3.7%)	41.1	(0.4%)	38.5	(6.6%)
Average Improvement	-	**5.3%**		**1.8%**		**10.7%**	

Table A.7: Word Error Rate - No Speaker Adaptation - 40k Vocabulary - "Search50"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	51.2	50.7	(1.0%)	50.3	(1.7%)	49.3	(3.7%)
Lecture 2	39.0	39.4	(-1.1%)	38.5	(1.2%)	38.2	(2.0%)
Lecture 3	38.3	33.4	(12.8%)	38.4	(-0.4%)	31.2	(18.6%)
Lecture 4	31.3	26.8	(14.4%)	31.3	(-0.1%)	25.5	(18.5%)
Lecture 5	31.6	30.6	(2.9%)	31.6	(0.0%)	28.6	(9.4%)
Lecture 6	46.1	45.1	(2.0%)	46.0	(0.2%)	44.7	(2.9%)
Lectures 1-6	41.3	39.7	(3.7%)	40.9	(0.8%)	38.6	(6.4%)
Average Improvement	-	**5.3%**		**0.4%**		**9.2%**	

Table A.8: Word Error Rate - No Speaker Adaptation - 40k Vocabulary - "Search500"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	50.4	51.5	(-2.0%)	49.2	(2.4%)	49.5	(1.9%)
Lecture 2	37.9	38.6	(-1.8%)	37.1	(2.0%)	37.5	(1.0%)
Lecture 3	38.4	33.5	(12.6%)	36.8	(4.1%)	30.6	(20.3%)
Lecture 4	29.9	25.4	(15.1%)	29.6	(0.8%)	24.2	(19.1%)
Lecture 5	29.3	31.9	(-8.8%)	24.9	(15.1%)	24.0	(18.2%)
Lecture 6	45.5	44.2	(2.9%)	45.0	(1.0%)	43.5	(4.3%)
Lectures 1-6	40.3	39.2	(2.8%)	39.5	(1.8%)	37.8	(6.2%)
Average Improvement	-	**3.0%**		**4.2%**		**10.8%**	

Table A.9: Word Error Rate - No Speaker Adaptation - 90k Vocabulary - "Search50"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	50.4	51.5	(-2.0%)	49.0	(2.9%)	49.5	(1.8%)
Lecture 2	37.9	38.6	(-1.8%)	37.2	(1.9%)	37.3	(1.6%)
Lecture 3	38.4	33.5	(12.6%)	36.8	(4.1%)	30.7	(19.9%)
Lecture 4	29.9	25.4	(15.1%)	29.3	(2.1%)	24.1	(19.5%)
Lecture 5	29.3	31.9	(-8.8%)	28.8	(1.9%)	25.5	(13.2%)
Lecture 6	45.5	44.2	(2.9%)	44.9	(1.4%)	43.4	(4.5%)
Lectures 1-6	40.3	39.2	(2.8%)	39.4	(2.2%)	37.7	(6.4%)
Average Improvement	-	**3.0%**		**2.4%**		**10.1%**	

Table A.10: Word Error Rate - No Speaker Adaptation - 90k Vocabulary - "Search500"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	50.4	50.5	(-0.3%)	49.9	(1.0%)	48.8	(3.2%)
Lecture 2	37.2	37.9	(-1.8%)	36.9	(0.7%)	36.9	(0.7%)
Lecture 3	37.3	33.0	(11.4%)	36.3	(2.6%)	30.0	(19.6%)
Lecture 4	25.4	24.8	(2.7%)	25.1	(1.3%)	23.4	(7.9%)
Lecture 5	29.9	29.7	(0.6%)	26.6	(11.1%)	21.8	(27.2%)
Lecture 6	43.8	43.3	(1.1%)	43.7	(0.1%)	42.6	(2.6%)
Lectures 1-6	38.6	38.4	(0.5%)	38.2	(0.9%)	37.1	(3.9%)
Average Improvement	-	**2.3%**		**2.8%**		**10.2%**	

Table A.11: Word Error Rate - No Speaker Adaptation - 300k Vocabulary - "Search50"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	50.4	50.5	(-0.3%)	49.1	(2.5%)	48.7	(3.2%)
Lecture 2	37.2	37.9	(-1.8%)	37.0	(0.6%)	37.1	(0.1%)
Lecture 3	37.3	33.0	(11.4%)	36.2	(2.8%)	30.4	(18.4%)
Lecture 4	25.4	24.8	(2.7%)	24.8	(2.6%)	23.2	(8.7%)
Lecture 5	29.9	29.7	(0.6%)	28.6	(4.3%)	25.1	(16.1%)
Lecture 6	43.8	43.3	(1.1%)	43.5	(0.5%)	42.8	(2.1%)
Lectures 1-6	38.6	38.4	(0.5%)	37.9	(1.6%)	37.2	(3.7%)
Average Improvement	-	**2.3%**		**2.2%**		**8.1%**	

Table A.12: Word Error Rate - No Speaker Adaptation - 300k Vocabulary - "Search500"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	43.1	42.4	(1.7%)	42.7	(1.0%)	41.0	(5.0%)
Lecture 2	34.9	35.7	(-2.3%)	34.3	(1.6%)	33.9	(2.7%)
Lecture 3	33.4	27.3	(18.2%)	34.7	(-3.8%)	27.5	(17.6%)
Lecture 4	28.3	23.9	(15.6%)	28.5	(-0.6%)	22.7	(20.0%)
Lecture 5	28.4	28.8	(-1.3%)	25.5	(10.4%)	21.2	(25.3%)
Lecture 6	37.4	36.4	(2.6%)	37.6	(-0.6%)	35.7	(4.4%)
Lectures 1-6	35.6	34.1	(4.2%)	35.4	(0.4%)	32.8	(7.9%)
Average Improvement	-	**5.8%**		**1.3%**		**12.5%**	

Table A.13: Word Error Rate - Speaker Adaptation - 40k Vocabulary - "Search50"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	43.1	42.4	(1.7%)	42.5	(1.5%)	40.8	(5.5%)
Lecture 2	34.9	35.7	(-2.3%)	34.7	(0.6%)	34.2	(2.0%)
Lecture 3	33.4	27.3	(18.2%)	33.1	(0.9%)	26.2	(21.6%)
Lecture 4	28.3	23.9	(15.6%)	28.6	(-0.8%)	22.5	(20.6%)
Lecture 5	28.4	28.8	(-1.3%)	26.2	(7.8%)	23.6	(16.9%)
Lecture 6	37.4	36.4	(2.6%)	37.5	(-0.5%)	35.6	(4.7%)
Lectures 1-6	35.6	34.1	(4.2%)	35.4	(0.4%)	32.7	(8.1%)
Average Improvement	-	**5.8%**		**1.6%**		**11.9%**	

Table A.14: Word Error Rate - Speaker Adaptation - 40k Vocabulary - "Search500"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	42.1	42.6	(-1.1%)	40.8	(3.2%)	40.3	(4.2%)
Lecture 2	33.7	34.5	(-2.3%)	32.8	(2.7%)	33.2	(1.5%)
Lecture 3	32.9	28.0	(14.9%)	33.2	(-0.7%)	26.6	(19.2%)
Lecture 4	26.8	22.3	(16.8%)	26.6	(0.7%)	21.2	(21.1%)
Lecture 5	25.5	27.7	(-8.7%)	20.7	(18.9%)	19.2	(24.6%)
Lecture 6	36.2	35.2	(2.7%)	36.7	(-1.5%)	34.6	(4.3%)
Lectures 1-6	34.4	33.2	(3.4%)	33.8	(1.6%)	31.7	(7.6%)
Average Improvement	-	**3.7%**		**3.9%**		**12.5%**	

Table A.15: Word Error Rate - Speaker Adaptation - 90k Vocabulary - "Search50"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	42.1	42.6	(-1.1%)	40.7	(3.2%)	40.5	(3.9%)
Lecture 2	33.7	34.5	(-2.3%)	33.0	(2.1%)	33.5	(0.7%)
Lecture 3	32.9	28.0	(14.9%)	33.4	(-1.3%)	25.6	(22.3%)
Lecture 4	26.8	22.3	(16.8%)	26.5	(1.2%)	21.3	(20.7%)
Lecture 5	25.5	27.7	(-8.7%)	21.4	(15.9%)	21.0	(17.4%)
Lecture 6	36.2	35.2	(2.7%)	36.1	(0.1%)	34.4	(4.9%)
Lectures 1-6	34.4	33.2	(3.4%)	33.7	(1.8%)	31.8	(7.3%)
Average Improvement	-	**3.7%**		**3.5%**		**11.7%**	

Table A.16: Word Error Rate - Speaker Adaptation - 90k Vocabulary - "Search500"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	42.1	42.3	(-0.4%)	41.3	(2.0%)	40.3	(4.4%)
Lecture 2	33.1	33.8	(-2.0%)	32.7	(1.2%)	32.8	(1.0%)
Lecture 3	33.1	27.6	(16.7%)	33.6	(-1.6%)	27.1	(18.0%)
Lecture 4	22.2	21.5	(2.9%)	21.9	(1.4%)	20.2	(9.0%)
Lecture 5	26.0	25.7	(1.4%)	20.7	(20.6%)	19.4	(25.5%)
Lecture 6	34.4	33.8	(1.7%)	34.4	(-0.1%)	33.4	(3.1%)
Lectures 1-6	32.7	32.4	(0.7%)	32.2	(1.3%)	31.1	(4.6%)
Average Improvement	-	**3.4%**		**3.9%**		**10.2%**	

Table A.17: Word Error Rate - Speaker Adaptation - 300k Vocabulary - "Search50"-Language Model

Word Error Rate (%)								
Adaptation	Baseline	Only Vocab		Only LM		Both Vocab & LM	
Lecture 1	42.1	42.3	(-0.4%)	40.8	(3.3%)	39.9	(5.3%)
Lecture 2	33.1	33.8	(-2.0%)	33.1	(0.2%)	33.1	(0.1%)
Lecture 3	33.1	27.6	(16.7%)	33.2	(-0.2%)	26.8	(19.1%)
Lecture 4	22.2	21.5	(2.9%)	21.7	(2.2%)	20.4	(8.0%)
Lecture 5	26.0	25.7	(1.4%)	21.4	(17.7%)	20.9	(19.8%)
Lecture 6	34.4	33.8	(1.7%)	34.3	(0.4%)	33.1	(3.8%)
Lectures 1-6	32.7	32.4	(0.7%)	32.1	(1.7%)	31.1	(4.7%)
Average Improvement	-	**3.4%**		**3.9%**		**9.4%**	

Table A.18: Word Error Rate - Speaker Adaptation - 300k Vocabulary - "Search500"-Language Model

Bibliography

[Asso99] I. P. Association. *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet.* Cambridge University Press. 1999. ISBN 978-0521637510.

[Brid06] R. Bridges. *On English Homophones.* Echo Library. 2006. ISBN 978-1847029133.

[ChGo98] S. Chen und J. Goodman. An empirical study of smoothing techniques for language modeling. Technischer Bericht, Technical report TR-10-98, Computer Science Group, Harvard University, August 1998.

[DaMe80] S. Davis und P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics Speech and Signal Processing* 28(4), 1980, S. 357–366.

[EiXu10] A. Eisele und J. Xu. Improving Machine Translation Performance Using Comparable Corpora. In *Proc. LREC*, Nr. May, 2010, S. 35–39.

[F07] C. Fügen, A. Waibel und M. Kolss. Simultaneous translation of lectures and speeches. *Machine translation* 21(4), November 2007, S. 209–252.

[GHCM+07] J. R. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh und R. Barzilay. Recent Progress in the MIT Spoken Lecture Processing Project. In *Proc. Interspeech*, 2007, S. 2553–2556.

[GHHW04] J. R. Glass, T. J. Hazen, L. Hetherington und C. Wang. Analysis and Processing of Lecture Audio Data: Preliminary Investigations. In *Proc. HLT-NAACL*, 2004, S. 9–12.

[HuAH01] X. Huang, A. Acero und H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall. 2001. ISBN 978-0130226167.

[JeMe80] F. Jelinek und R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In E Gelsema und L Kanal (Hrsg.), *Pattern Recognition in Practice*, S. 381–397. North Holland, 1980.

[Jone04] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 60(5), 2004, S. 493–502.

[KaNA08] T. Kawahara, Y. Nemoto und Y. Akita. Automatic Lecture Transcription by Exploiting Presentation Slide Information for Language Model Adaptation. In *Proc. ICASSP*, 2008, S. 4929–4932.

[Kawa10] T. Kawahara. Automatic Transcription of Parliamentary Meetings and Classroom Lectures. In *Proc. ISCSLP*, 2010, S. 1–6.

[KnNe95] R. Kneser und H. Ney. Improved backing-off for m-gram language modeling. In *Proc. ICASSP*, Band 1. Ieee, 1995, S. 181–184.

[Koeh10] P. Koehn. *Statistical Machine Translation.* Cambridge University Press. 2010. ISBN 978-0521874151.

[KoOM03] P. Koehn, F. J. Och und D. Marcu. Statistical phrase-based translation. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology NAACL 03* 1(June), 2003, S. 48–54.

[KuLe51] S. Kullback und R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics* 22(1), 1951, S. 79–86.

[KWKN+08] M. Kolss, M. Wölfel, F. Kraft, J. Niehues, M. Paulik und A. Waibel. Simultaneous German-English Lecture Translation. In *Proc. IWSLT*, 2008, S. 174–181.

[LBGA+09] L. Lamel, E. Bilinski, J.-L. Gauvain, G. Adda, C. Barras und X. Zhu. The LIMSI RT07 Lecture Transcription System. *Multimodal Technologies for Perception of Humans*, 2009, S. 442–449.

[MaLW11] P. Maergner, I. Lane und A. Waibel. Unsupervised Vocabulary Selection for Domain-Independent Simultaneous Lecture Translation. In *Proc. MT Summit XIII*, Xiamen, China, 2011. S. 89–96.

[MaTN08] C. Martins, A. Teixeira und J. Neto. Dynamic language modeling for a daily broadcast news transcription system. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on.* IEEE, 2008, S. 165–170.

[MiSA11] J.-B. Michel, Y. K. Shen und A. P. Aiden. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science* Band 331, Dezember 2011, S. 176–182.

[MKLW11] P. Maergner, K. Kilgour, I. Lane und A. Waibel. Unsupervised Vocabulary Selection for Simultaneous Lecture Translation. In *Proc. IWSLT*, San Francisco, USA, 2011. S. 214–221.

[MuPB07] C. Munteanu, G. Penn und R. Baecker. Web-based Language Modelling for Automatic Lecture Transcription. In *Proc. Interspeech*, Nr. August, 2007, S. 2353–2356.

[NeEK94] H. Ney, U. Essen und R. Kneser. On structuring probabilistic dependences in stochastic language modeling. *Computer Speech and Language* 8(1), 1994, S. 1–38.

[OlPL05] D. Olszewski, F. Prasetyo und K. Linhard. Steerable Highly Directional Audio Beam Loudspeaker. In *Proc. Interspeech*, 2005, S. 137–140.

[Rose96] R. Rosenfeld. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer, Speech and Language* Band 10, 1996, S. 187–228.

[SaBu88] G. Salton und C. Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management* 24(5), 1988, S. 513–523.

[SMFW01] H. Soltau, F. Metze, C. Fügen und A. Waibel. A one-pass decoder based on polymorphic linguistic context assignment. In *Proc. ASRU*, 2001, S. 214–217.

[StKN10] S. Stüker, K. Kilgour und J. Niehues. Quaero Speech-to-Text and Text Translation Evaluation Systems. In *Proc. HLRS*. Springer, 2010, S. 529–542.

[Stol02] A. Stolcke. SRILM-An Extensible Language Modeling Toolkit. In *Proc. ICSLP*, Band 2. Citeseer, 2002, S. 901–904.

[TaBC98] P. Taylor, A. W. Black und R. Caley. The architecture of the Festival speech synthesis system. In *The Third ESCA Workshop in Speech Synthesis*, Jenolan Caves, Australia, 1998. Citeseer, S. 147–151.

[VeWa03] A. Venkataraman und W. Wang. Techniques for effective vocabulary selection. *Proc. of the 8th European Conference on Speech Communication and Technology*, Juni 2003, S. 4.

[VoNT96] S. Vogel, H. Ney und C. Tillmann. HMM-based word alignment in statistical translation. *Proceedings of the 16th conference on Computational linguistics* Band 96pp, 1996, S. 836–841.

[VZHT+03] S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venugopal, B. Zhao und A. Waibel. The CMU statistical machine translation system. In *Proc. MT Summit IX*, Band 9, 2003, S. 54.

[WaF8] A. Waibel und C. Fügen. Spoken Language Translation. *IEEE Signal Processing Magazine* 25(3), Mai 2008, S. 70–79.

[WiBe91] I. H. Witten und T. C. Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory* 37(4), 1991, S. 1085–1094.

[YISF+07] H. Yamazaki, K. Iwano, K. Shinoda, S. Furui und H. Yokota. Dynamic Language Model Adaptation Using Presentation Slides for Lecture Speech Recognition. In *Proc. Interspeech*, 2007, S. 2349–2352.