A System for Simultaneous Translation of Lectures and Speeches

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der Fakultät für Informatik
der Universität Fridericiana zu Karlsruhe (TH)

genehmigte

Dissertation

von

Christian Fügen

aus Mannheim

Tag der mündlichen Prüfung:	07. November 2008
Erster Gutachter:	Prof. Dr. Alexander Waibel
Zweiter Gutachter:	Prof. Dr. Tanja Schultz

Abstract

This thesis realizes the first existing automatic system for simultaneous speech-to-speech translation. The focus of this system is the automatic translation of (technical oriented) lectures and speeches from English to Spanish, but the different aspects described in this thesis will also be helpful for developing simultaneous translation systems for other domains or languages.

With growing globalization, information exchange between people from different points of origin increases in importance. In the case of the European Union or the United Nations often Arabic, Chinese, English, Russian, Spanish or French is used as common communication language, but not all people are able to speak fluently in these languages. Therefore, often simultaneous or consecutive interpretations are necessary to overcome this language barrier. But the costs for such interpretation services are increasing continuously — about 1 billion Euros are spent per year within the European Union.

Hence, it is not surprisingly that the governmental funding of research in the domain of spoken language translation is increasing. Large research projects have been launched like TC-STAR in the EU and GALE in the USA. In contrast to the system proposed in this thesis, the main focus is to achieve high quality translation of text and speech wherefore systems are required which run at multiples of real-time.

This thesis deals with the question on how a simultaneous translation system can be built and determines whether satisfactory quality can be achieved with state-of-the-art components. A main focus is to increase the performance of the speech recognizer and the interface between the speech recognition and machine translation components.

It will be shown how the performance can be increased by using speaker adaptation techniques. With an amount of 15 minutes of speech, the error rate was increased by 5.6% relative using supervised adaptation techniques and by 2.1% using unsupervised adaptation techniques. Furthermore, the importance of online adaptation during decoding was shown.

Since topics between lectures and speeches may greatly vary, a domain adaptation framework is proposed, which is able to automatically adapt the system towards a new domain by using language model adaptation.

The adaptation level and expected performance improvement from 3-4% relative in WER and 12-22% relative in BLEU depends on the information available prior to the presentation. Solutions are proposed for information ranging from the speakers name to past publications of the speaker through to the presentation slides. Relevant data for language model adaptation was collected by querying a search engine and retrieving the web pages which were returned as the result of the query. A $tf - idf$ based heuristic was developed for generating suitable queries.

Besides recognition and translation quality, high speed and short latency are important for a simultaneous translation system. Therefore, speed-up techniques like search-space pruning and Gaussian selection are explored. To reduce the latency, a streaming approach was implemented, in which the recognizer returns steadily partial hypotheses for a continuous input stream of speech. A resegmentation component was introduced to chunk the continuous stream of partial hypotheses in semantic segments; short enough to keep the latency low, but long enough to not degrade translation quality. By using the proposed techniques, decoding speed could be reduced by 27% to a real-time factor of 0.78 and latency could be reduced to 2-3 seconds, both with only minor decrease in translation quality.

For delivering the output of the simultaneous translation system to the audience several promising technologies such as targeted audio loudspeakers will be explored.

Compared to a human interpreter, the automatic system was judged in a human end-to-end evaluation in the category of fluency to 2.3, and the interpreter to 3 on a scale ranging from 1 (bad) to 6 (very good). Furthermore, with the help of an questionnaire, it could be shown that about 50% of the questions could be answered by the judges in case of the automatic system and about 75% in case of the human-interpreter.

Kurzzusammenfassung

Die vorliegende Arbeit realisiert das erste existierende automatische Übersetzungssystem, das für die Simultanübersetzung von (technischen) Vorträgen oder Reden von Englisch nach Spanisch geeignet ist.

Die zunehmende Globalisierung erfordert den Fluss von Information zwischen Personen unterschiedlicher Herkunft und Muttersprache. Beispielsweise besteht die Europäische Union aus 27 verschiedenen Staaten und die Vereinten Nationen ist ein Zusammenschluss von sogar 192 Staaten. Zwar dient oft Arabisch, Chinesisch, Englisch, Russisch, Spanisch oder Französisch als Kommunikations- bzw. Amtssprache, jedoch werden diese Sprachen nicht von allen gleichermaßen gut gesprochen. Gerade jedoch in wichtigen Gesprächen, Debatten, oder Verhandlungen möchte kaum jemand darauf verzichten diese in der eigenen Muttersprache zu führen, in der er sich am sichersten fühlt. Insofern werden z.B. im Europäischen Parlament alle Debatten simultan in zur Zeit 23 Amtssprachen interpretiert – ein erheblicher Kostenfaktor. Für kleinere Veranstaltungen wie z.B. Forschungskonferenzen sind solche Kosten nicht tragbar, weshalb man davon ausgehen kann, dass manche Vorträge aufgrund dieser Kommunikationsbarriere einfach nicht stattfinden. In den USA ist Sprachübersetzung vor allem bei der Prävention von Terroranschlägen und aufgrund der Konflikte mit anderen Ländern wie dem Irak immens wichtig geworden. Da jedoch die Datenmengen, die über Fernsehstationen oder im Internet in fremden Sprachen verfügbar gemacht werden, riesig sind, sind diese nur noch durch automatische Methoden analysierbar.

Insofern ist es nicht überraschend, dass gerade in letzter Zeit zunehmend Forschungsgelder in Sprachübersetzungsprojekte in der EU (TC-STAR) und in den USA (GALE) investiert wurden. Das Ziel solcher Projekte, ist es auf großen Domänen eine höchstmögliche Sprachübersetzungsqualität zu erlangen. Deshalb besitzen solche Systeme Verarbeitungsgeschwindigkeiten von mehreren zig Echtzeitfaktoren. Es gibt jedoch auch Systeme, die sehr viel kürzere Antwortzeiten haben und sogar schon auf mobilen Plattformen funktionieren, sich jedoch aber meist nur auf kleine Domänen, wie z.B. Terminabsprachen oder touristische Phrasen beschränken.

Mit dem in dieser Arbeit vorgestelltem Simultanübersetzers lässt sich nun erstmals die Kommunikationsbarriere auch in kleineren Veranstaltungen wie z.B. Vorlesungen an Universitäten überwinden. Da die Performance,

d.h. die Übersetzungsqualität und die Latenz zwischen Vortragendem und Übersetzung von größter Bedeutung sind, beschäftigt sich diese Arbeit mit den Problemen beim Aufbau eines solchen Systems und deren Lösungen.

Sprecheradaption

Es wird gezeigt, wie stark sich die Performance des Systems durch verschiedene Ansätze zur überwachten und unüberwachten Sprecheradaption verbessern lässt. Bei einer verfügbaren Datenmenge von etwa 15 Minuten, konnte die Fehlerrate um 5.6% durch überwachte und immerhin noch um 2.1% durch unüberwachte Sprecheradaption reduziert werden. Es konnte auch die Wichtigkeit einer fortlaufenden Adaption während des Vortrags gezeigt werden.

Domänenadaption

Es werden verschiedene Ansätze zur Domänenadaption in Abhängigkeit der zur Verfügung stehenden Adaptionsdaten untersucht. Angefangen mit dem Namen des Vortragenden, über mehr oder weniger verwandte Publikationen bis hin zu den Vortragsfolien wird gezeigt wie solche Informationen effektiv genutzt werden können. Hierzu wurde ein Framework entwickelt, in dem in Abhängigkeit der zur Verfügung stehenden Information ähnliche Daten aus dem Internet geladen werden, um damit automatisch die Sprachmodelle von Spracherkennung und Sprachübersetzung zu adaptieren. Um relevante Webseiten mit Hilfe von Suchmaschinen wie Google zu finden, wurde eine $tf - idf$ basierte Heuristik zur Generierung der Anfragen entwickelt. Es konnte gezeigt werden, dass die Webseiten, die mit Hilfe dieser Heuristik gesammelt wurden, themenverwandte Informationen enthalten. In Abhängigkeit des Hintergrundsprachmodells konnte durch die Adaption die Fehlerrate der Spracherkennung um 3-4% und der BLEU-Score der Sprachübersetzung um 12-22% verbessert werden. Ferner wurde untersucht, inwieweit sich auch das Vokabular des Spracherkennungssystem mit Hilfe dieser Daten auf die neue Domäne adaptiert werden kann.

Geschwindigkeit und Latenz

Im Gegensatz zu anderen Arbeiten im Bereich der Sprachübersetzung ist das in dieser Arbeit vorgestellte System das erste, das auch in Echtzeit in größeren Domänen wie Vorträge und Reden arbeitet. Insofern beschäftigt sich diese Arbeit auch mit dem Einfluss von verschiedenen Parametern wie Modellgröße (akustisches Modell, Sprachmodell), Suchraumbeschränkung (Pruning), und anderen Beschleunigungstechniken auf die Geschwindigkeit und Qualität von Spracherkennung und Sprachübersetzung. Neben der Geschwindigkeit ist auch eine geringe Latenz sehr wichtig, da sie die Kommunikation zwischen Publikum und Vortragendem aber auch innerhalb des

Publikums beeinflusst. Die Latenz entsteht durch die Serialisierung in der Abarbeitung der Eingaben, da der Sprachübersetzung möglichst semantisch abgeschlossene Einheiten übermittelt werden müssen, um eine gute Übersetzungsqualität zu gewährleisten. Insofern wird in dieser Arbeit gezeigt, wie eine solche Schnittstelle zwischen Spracherkennung und Sprachübersetzung realisiert werden kann und wie dadurch die Übersetzungsergebnisse beeinflusst werden. Die Geschwindigkeit des Spracherkenners konnte um 27% auf einen Echtzeitfaktor von 0.78 bei einer Verschlechterung der Fehlerrate von nur 2% reduziert werden. Ferner konnte die Latenz des Simultanübersetzers auf 2-3 Sekunden mit nur geringen Einbußen in der Übersetzungsqualität reduziert werden.

System und Übertragungsmedien

Des Weiteren wird die in dieser Arbeit entwickelte Gesamtarchitektur des Systems vorgestellt und verschiedene Übertragungsmedien im Hinblick auf ihre Eignung in verschiedenen Szenarien analysiert. Verschiedene vielversprechende Technologien wie z.B. Übersetzungsbrillen und gerichtete Ultraschalllautsprecher werden im Detail beschrieben.

Humanevaluation

Da es im allgemeinen sehr schwierig ist, die Qualität des Simultanübersetzers mit Hilfe von automatisch berechenbaren Gütemaßen zu beurteilen, wurde eine Humanevaluation durchgeführt. Die beiden wichtigsten Kriterien hierbei waren syntaktische Korrektheit, d.h. Flüssigkeit und semantische Korrektheit, d.h. Eignung der Übersetzung. Mit Hilfe eines Fragebogens und im Vergleich mit einem menschlichen Interpreter konnte gezeigt werden, dass mit Hilfe der automatischen Simultanübersetzung über 50% aller Fragen beantwortet werden konnten, während es bei der menschlichen Interpretation 75% waren. Die Flüssigkeit der Übersetzung wurde auf einer Skala von 1 (schlecht) bis 6 (besser) im Falle des menschlichen Interpreters mit einer 3 und im Falle des automatischen Systems mit einer 2.3 bewertet. Zusammenfassend ist zu sagen, dass gerade bei technisch anspruchsvollen Vorträgen auch menschliche Simultandolmetscher nicht in der Lage sind, diese korrekt zu übersetzen und dass ein automatisches System mindestens in der Lage ist dem Zuhörer den Kontext des Vortrags zu vermitteln – oftmals schon ausreichend, um jemandem genug Wissen zu vermitteln, der die Sprache des Vortragenden nicht versteht.

Danksagung

Zunächst möchte ich meinem Doktorvater Professor Dr. Alexander Waibel danken, der mir die Möglichkeit gab an den “Interactive Systems Laboratories” (ISL) zu arbeiten. Ganz besonders aber danken möchte ich ihm für sein Interesse, sein in mich gesetztes Vertrauen und den steten Ansporn diese Dissertation zu vollenden. Er konnte mich begeistern, nach meiner Tätigkeit als wissenschaftlicher Mitarbeiter, weiterhin für ihn zu arbeiten, um neue interessante Wege zu gehen und mit dabei zu sein, Spracherkennung und Sprachübersetzung einer größeren Masse näher zu bringen. Rückblickend kann ich sagen, dass die Zeit bei den ISL sehr viel Spaß gemacht hat und immer neue, interessante Herausforderungen bot.

Danken möchte ich auch Professor Dr. Tanja Schultz für die Übernahme des Korreferats, für interessante Diskussionen und für hilfreiche Kommentare – auch unter Zeitdruck – zu dieser Arbeit. Ein besonderes Dankeschön geht auch an Kornel Laskowski, der trotz seiner beschränkten zur Verfügung stehenden Zeit, bereit war, diese Arbeit Korrektur zu lesen und mich so sicherlich vor vielen Peinlichkeiten im Englischen bewahrte.

Früh konnten mich die Mitarbeiter der ISL für die Thematik der automatischen Spracherkennung begeistern. Vor dem Beginn meines Studiums, während der “O-Phase”, wurden wir durch verschiedene Lehrstühle geführt. Ivica Rogina und Monika Woszczyna führten dabei Spracherkennung und eines der ersten C-Star Übersetzungssysteme vor. Bald nach dem Vordiplom begann ich dann auch an den ISL als wissenschaftliche Hilfskraft (HiWi), und schrieb dort auch meine Studienarbeit und Diplomarbeit. Ein ganz besondere Dank hierbei gilt Ivica Rogina, der mich während meiner ganzen Zeit als HiWi bis hin zur Diplomarbeit betreute, und mir mit seinem Fachwissen und Rat immer zur Seite stand. Soweit ich mich erinnern kann, kam es nie vor, dass er sich für eine Frage meinerseits nicht sofort Zeit genommen hätte.

Vielen Dank an Martin Westphal, meinem ersten Zimmerkollegen, von dem ich relativ schnell die Verantwortung für LingWear übertragen bekommen habe. Meinen langjährigen Zimmerkollegen Florian Metze und Hagen Soltau gilt mein besonderer Dank. Als “Ibis Gang” haben wir viel gemeinsam Unternommen und es war ein tolles Erlebnis, den neuen Ibis Decoder endlich am Laufen zu sehen und mit ihm weiterhin arbeiten zu dürfen. Ein

Dankeschön auch an Thilo Köhler und Kay Rottmann, meinen jetzigen Zimmerkollegen – auch wenn sie nicht immer daran denken meine Blumen zu gießen.

Muntsin Kolss möchte ich ganz besonders danken. Ohne ihn wäre diese Arbeit nicht zustande gekommen. Er war in erster Linie für das Training des in dieser Arbeit verwendeten Sprachübersetzungssystem verantwortlich. Ihm zur Seite stand zeitweise auch Matthias Paulik, der vor allem durch eine sorgfältige und zeitraubende Aufbereitung der parallelen Trainingsdaten und des Alignments das Übersetzungssystem signifikant verbessern konnte. In diesem Zusammenhang möchte ich auch Dietmar Bernreuther für das Training des ersten Englisch-Spanisch Übersetzungssystem und Stephan Vogel für seine Unterstützung in Fragen zur Übersetzung danken. Ein herzliches Dankeschön auch an Sebastian Stüker, für die vielen inspirierenden Gespräche die wir miteinander hatten aber vor allem für seine stetige Hilfsbereitschaft und Engagement durch die er mir manche Demo des Lecture Translation Systems erleichterte oder sogar komplett abnahm.

Danken möchte ich auch den anderen ehemaligen und aktuellen Mitarbeitern der ISL wie Keni Bernadin, Matthias Denecke, Hazim Ekenel, Michael Finke, Jürgen Fritsch, Petra Gieselmann, Hermann Hild, Hartwig Holzapfel, Shajith Ikbal, Thomas Kemp, Florian Kraft, Kenichi Kumatani, John McDonough, Kai Nickel, Klaus Ries, Tanja Schultz, Thomas Schaaf, Rainer Stiefelhagen, Matthias Wölfel, Monika Woszczyna – und ich hoffe, ich habe keinen vergessen. Ein Dank auch an Mari Ostendorf für ihre interessanten Vorträge während ihrer Zeit als Gast an den ISL.

Bei meinen Reisen an die Carnegie Mellon University in Pittsburgh durfte ich auch die Mitarbeiter dort persönlich kennen lernen, mit denen wir sonst immer nur per E-Mail kommunizierten. Mein besonderer Dank gilt hier Susi Burger, die immer bereit war mir in dieser Zeit Unterkunft zu gewähren. Mit ihrem Team an Studenten und ihrem persönlichen Einsatz hat sie in vielen Stunden, die in dieser Arbeit verwendeten Daten transkribiert und übersetzt.

Bedanken möchte ich mich bei Frank Dreilich, Norbert Berger und Dan Valsan für die Sicherstellung des Rechnerbetriebes. Ein ganz besonderes Dankeschön gilt auch dem Sekretariat, vor allem Silke Dannenmaier und Annette Römer, aber auch an Margit Rödder für die Übernahme vieler organisatorischer Arbeiten an den ISL.

Schließlich möchte ich mich bei meinen Eltern bedanken, die es mir ermöglichten, mich so zu entfalten wie ich es wollte. Mit ihrer stetigen Hilfsbereitschaft und Unterstützung gaben sie mir immer Rückhalt, auch in schwierigeren Zeiten. Auch meinem Bruder gilt Dank, der momentan selber dabei ist seine Dissertation abzuschließen. Ihm wünsche ich dabei viel Erfolg.

Bedanken möchte ich mich ganz besonders bei meiner Ehefrau Sani für die Liebe und Kraft, die sie mir gegeben hat. Ohne sie wäre meine Promotion nicht möglich gewesen. In all den Jahren an meiner Seite hat sie den Stress

und die Belastung mit mir geteilt und getragen. Auch möchte ich ihr dafür danken, dass sie für unsere Kinder Jonas und Svea immer eine liebevolle und engagierte Mutter ist. Sicherlich war es für sie nicht ganz einfach den Kindern oft auch den Vater zu ersetzen.

> Diese Arbeit widme ich meinen Kindern und wünsche, dass sie niemals im Leben ihre Wissbegierde und ihr Durchhaltevermögen verlieren werden.

Mannheim, 21. September 2008

Christian Fügen

Contents

Chapter 1

Introduction and Motivation

Estimates for the number of existing languages today range from 4000 to 6000. At the same time, the phenomenon of globalization requires an active flow of information among people speaking a wide variety of languages. Lectures are an effective way of performing this dissemination. Personal talks are preferable over written publications because they allow the speaker to tailor his or her presentation to the needs of a specific audience, and at the same time allow the listeners to access information relevant for them through interaction with the speaker. Currently, many lectures simply do not take place because no matter how intensively one studies a foreign language, one will always be more expressive, more fluent, and more precise in one's native tongue, and human translators are too expensive. The use of modern machine translation techniques can potentially provide affordable translation services to a wide audience, making it possible to overcome the language barrier for almost everyone.

So far, speech translation research has focused on limited domains, such as the scheduling of meetings, basic tourist expressions, or the pre-arrival reservation of hotel rooms. The development of these recognition and translation systems has happened in phases. At first, only single, isolated phrases could be recognized and translated. The phrases had to be spoken in a clean and controlled manner adhering to a predetermined grammar, and only previously seen phrases could be translated. In the next phases, the restrictions on the speaking style were lifted, leading to the emergence of recognition systems for conversational and spontaneous speech. At the same time, the allowed discourse in terms of vocabulary and sentences increased. Large vocabulary continuous speech recognition systems became reality. Similar developments have been observed in the field of machine translation.

This thesis realizes the first existing automatic system for simultaneous speech-to-speech translation. The focus of this system is the automatic translation of lectures and speeches from English to Spanish, but the different aspects described in this thesis will also be helpful for developing simultaneous translation systems for other domains or languages. Several different components including automatic speech recognition (ASR), machine translation (MT), and text-to-speech synthesis (TTS) are involved in such a system, and this thesis examines end-to-end performance requirements and how they can be met. Two performance aspects are of particular interest: translation quality and system latency. Both performance aspects rely on the performance of the sub-components and their interaction with one another.

To improve the performance of speech recognition or machine translation, system adaptation is the most common technique. This work investigates the different levels of topic adaptation for speech recognition and machine translation, dependent on the amount and type of data available prior to a specific lecture. Possibilities are the speaker's name, more or less related research papers up to presentation slides. It is shown how this information can be effectively used to improve the performance of the system. This work also demonstrates how the performance of the system can be improved by supervised or unsupervised speaker adaptation.

In contrast to other work focusing on speech-to-speech translation, this system is the first one which operates in real-time. Typically, speech recognition is performed in several consecutive steps of decoding and unsupervised adaptation followed by sentence-based machine translation to achieve the best possible translation quality. However, this is accompanied by a major increase in real-time and therefore unsuitable for simultaneous translation. An important aspect of this thesis is its focus on an analysis of model size, pruning parameter, or other speed-up techniques influencing the processing speed of speech recognition and machine translation.

In addition to processing speed, latency is also an important attribute of real-time systems. The reason of this is that the inter-communication between the lecturer and the audience or between people in the audience is negatively affected, if the latency is to high. This work demonstrates how a low-latency interface between speech recognition and machine translation can be designed, and how latency-related problems occurring especially during machine translation can be solved.

Since it is difficult to judge the quality of such a system using exclusively automatic measures, the translation quality of the automatic system was evaluated by humans and compared with the quality of a human interpreter of the same lectures.

Last but not least, this thesis explores translation delivery in ways other than using traditional head-phones. Since head-phones hinder communication between people in the audience, possibilities were explored of delivering a target oriented translation, but without disturbing other people in the audience. Several innovative technologies such as heads-up display goggles or beam-steered ultrasound loudspeakers will be described in more detail and compared. Note that although a TTS component is essential in a simultaneous translation system, the development and optimization of such a component is not part of this thesis. Instead, a TTS system from Cepstral[1] was used.

The remainder of this chapter presents the goals and contributions of this work, and gives an overview of the structure of this thesis.

1.1 Goals

The major goal of this thesis is easily formulated: To develop an automatic speech-to-speech translation system which is able to simultaneously translate lectures and speeches at satisfactory quality. But what is a satisfactory translation quality?

To answer this question, possible applications of the proposed system have to be defined. While specialized automatic systems exist for text translation, able to achieve reasonable translation results in a specific domain, it is clear that current automatic simultaneous translation systems are unable to achieve the same performance as a human interpreter. But in the author's opinion an automatic system become useful the moment that if people not understanding the language of the speaker at all are at least able to understand the rough content of the speech or lecture correctly. This means that, in situations where a human interpretation is simply too costly, automatic translation systems may be preferred. For the purpose of this thesis, "satisfactory quality" is achieved when the rough content of a lecture or speech is correctly transferred.

A goal of this thesis is to determine whether satisfactory quality can be achieved with current state-of-the art technologies, and to what extent.

Another goal of this thesis is to explain the problems involved in building such a system and to identify and describe several solutions to them.

1.2 Outline

This work can be divided into three parts. The first part comprises Chapters 2 to 4 and compares the advantages and disadvantages of human simultaneous interpretation with those of automatic simultaneous translation. Furthermore, it introduces the lecture scenario and a first baseline system. In the second part, Chapters 5 to 8, the main issues of a simultaneous translation system are discussed, namely speaker adaptation, topic adaptation, latency and real-time, as well as the chunking of the speech recognizer's hypotheses for optimal use in machine translation. Although some of the developed techniques are applied to the machine translation as well, the main focus of this thesis is improving the speech recognition. The simultaneous translation system itself together with some delivery aspects as well

[1] http://www.cepstral.com

as its end-to-end evaluation is presented in the third part, Chapters 9 and 10.

More specifically, Chapter 2 clarifies the differences between the term *Translation* and *Interpretation* and describes the challenges in human interpretation. In contrast thereto, Chapter 3 points out the advantages of an automatic simultaneous translation system compared to human interpreters and defines some demands on an automatic translation system. In addition, the application scenario, i.e. lectures on which this thesis focus is introduced.

Chapter 4 introduces the data available for system training, i.e. acoustic model, language model, and translation model training, and describes the lecture data used as development and evaluation sets. Furthermore, the speech recognition and machine translation systems used as a baseline for the experiments in the following Chapters are introduced.

The next Chapter 5 deals with speaker adaptation in the simultaneous translation system. First, the adaptation techniques used are introduced and the differences between online and offline as well as supervised and unsupervised adaptation are described. After this, the results achieved by using the introduced techniques and applying them to the acoustic model of the speech recognizer are presented.

In Chapter 6 a framework for topic adaptation is introduced. Depending on the information available for a particular talk or speaker, different levels of adaptation can be applied to a language model. For language model adaptation a topic dependent adaptation schema is presented, which base on linear language model interpolation with components build on relevant data retrieved from the Internet. Therefore, a *tf-idf* based method will be proposed, which extracts topic related queries out of the given data for querying a search engine.

The focus of Chapter 7 are the latency and real-time issues of the simultaneous translation system. First, the search space pruning within the used speech recognizer, *Ibis* is analyzed and after that the performance of different Gaussian selection techniques are compared. While the search space pruning and Gaussian selection are mainly responsible for reducing the decoding speed, the necessary changes for a standard speech recognizer to reducing the latency will be explained as well.

Chapter 8 concentrates on the interface between speech recognition and machine translation and deals with the question how a continuous stream of words delivered from the speech recognition can be optimally segmented in order to keep the latency of the simultaneous translation system low but the translation quality high. Therefore, an algorithm will be presented which tries to identify semantic boundaries.

The developed prototype of a system for simultaneous translation will be presented in Chapter 9 in more detail. Furthermore, it is reflected about different output or delivery technologies for the system.

Chapter 10 presents the results of the end-to-end evaluation. Besides an automatic evaluation also a human end-to-end evaluation was carried out. Chapter 11 concludes the work. The questionnaires used for the human end-to-end evaluation are presented in Appendix A.

This thesis will not give an introduction to the fundamentals of speech recognition and machine translation. Instead, readers not familiar with signal processing, acoustic modeling using Hidden Markov Models, statistical language modeling (LM), or statistical machine translation (SMT) are referred to [HAH01, SK06] for speech recognition and [HS92, Tru99] for machine translation.

Chapter 2

Human Simultaneous Interpretation

Everybody who speaks at least two languages knows that translation and especially simultaneous interpretation are very challenging tasks. One has to cope with the special nature of different languages such as terminology and compound words, idioms, dialect terms or neologisms, unexplained acronyms or abbreviations and proper names, but also stylistic differences and differences in the use of punctuation between two languages. Translation or interpretation is not a word-by-word rendition of what was said or written in a source language; instead, the meaning and intention of a given sentence has to be transferred in a natural and fluent way.

In this chapter, the differences between the terms *Interpretation* and *Translation*, especially in the context of this thesis, namely automatic simultaneous translation, are clarified in Section 2.1. Section 2.2 presents the world's largest employer for translators and interpreters, the European Commission, and the costs incurred by their services. Section 2.3 concentrates on the challenges in human interpretation and describes some techniques and compensatory strategies used by interpreters, as well as some factors and stylistic aspects responsible for the quality of simultaneous interpretation.

2.1 The Differences between Interpreting and Translating

Although the terms translation and interpretation are used interchangeably in everyday speech, they vary greatly in meaning. Both refer to the transference of meaning between two languages; however, *translation* refers to the transference of meaning from text to text with time and access to resources such as dictionaries, glossaries, et cetera. On the other hand *interpreting* is the intellectual activity that consists of facilitating oral or sign language communication between two or among three or more speakers who are not

Figure 2.1: Translation booths in the European Parliament's hemicycle at Brussels (from [Wik07a, Eur]).

speaking the same language. [Wik07b]

Both *interpreting* and *interpretation* can be used to refer to this activity, but the word *interpreting* is commonly used in avoiding the other meanings of the word *interpretation*.

The practitioner who orally translates for parties conversing in different languages or in sign language is called an *interpreter*. Interpreters must convey not only all elements of meaning, but also the intentions and feelings of the original, source language speaker. In fact, the end result is an intermediate stage of spoken communication, which aims to allow target language listeners to hear, perceive, and experience the message in a way that is as close as possible to the experience of those who understand the original, source language. [Wik07b]

Translators and interpreters are trained in entirely different manners. Translators receive extensive practice with representative texts in various subject areas, learn to compile and manage glossaries of relevant terminology, and master the use of both current document-related software (for example word processors, desktop publishing systems, and graphics or presentation software) and computer-assisted translation software tools. Interpreters, by contrast, are trained in precise listening skills under taxing conditions, memory and note-taking techniques for consecutive interpreting, and split-attention for simultaneous interpreting. [Wik07c]

The industry expects interpreters to be more than 80% accurate; that is to say that interpretation is an approximate version of the original. By contrast, translations should be over 99% accurate. [Wik07c]

2.1.1 Simultaneous and Consecutive Interpreting

There are two modes of interpretation: *simultaneous* and *consecutive*.

Simultaneous interpreting: In simultaneous interpreting, the interpretation occurs while the source language speaker speaks, as quickly as the interpreter can formulate the spoken message in the target language. At the European Parliament, for example, simultaneous interpretation occurs while the interpreter sits in a sound-proof booth, while listening with earphones to the speaker's source language message (see Figure 2.1). The interpreter then relays the message in the target language into a microphone to the target language listeners. Simultaneous interpreting is the most common mode used by sign language interpreters, as there is no audible language interference while both languages are being expressed simultaneously.

Consecutive interpreting: In consecutive interpretation, the interpreter speaks after the source-language speaker has finished speaking; the speech may be divided into sections. The interpreter is listening and taking notes as the speaker progresses. When the speaker finishes speaking or pauses, the interpreter consecutively renders the message in the target language, in its entirety, as though he or she were making the original speech. Frequently, an experienced consecutive interpreter prefers interpreting phrase by phrase, or shorter sentence portions, so as to approximate simultaneous interpretation. Because of this strategy, consecutive interpretation allows the full meaning to be understood before the interpreter renders the message into the target language. This often affords a more accurate and fully accessible interpretation than simultaneous interpreting.

2.1.2 Simultaneous Translation

Simultaneous interpreting sometimes is incorrectly referred to as *simultaneous translation* and the interpreter as the *translator*. However, in computer science, the terms *machine translation* (MT) or *automatic translation* are commonly used for systems translating text or speech from one language to another. The reason for that is that in the past, the main focus of machine translation was the translation of text, and spoken language translation (SLT) is only recently attracting a wider interest. Furthermore, the techniques used for text translation are almost identical to those used for spoken language translation nowadays. Therefore, throughout this thesis, the terms *simultaneous speech translation* or simply *simultaneous translation* are used for the automatic interpretation of spoken language.

2.2 Translating and Interpreting for the European Commission

The majority of interpreters work for international organizations like the United Nations, the European Union, or the African Union, whereas the world's largest employer of translators and interpreters is currently the European Commission (EC), with its two Directorate Generals for Translation[1]

[1] http://europa.eu.int/comm/dgs/translation/index_en.htm

and Interpretation[2].

The Directorate General for Translation (DGT) mainly provides translations of written text in and out of the 23 official languages of the European Union. There are more than 1800 translators working full-time on translating documents and on other language-related tasks, accompanied by some 600 support staff. In 2006, the DGT translated more than 1.5 million pages; 72% of the original texts were drafted in English, 14% in French, 2.7% in German, and 10.8% in the other 20 EU languages. English and French predominate, because they are the principal drafting languages in the European Commission. [Dir07b]

To support the translators, information technology, such as translation memory and machine translation technology, is often used. With translation memory technology, translators can avoid re-translating what has already been translated. At present, the central translation memory contains more than 84 million phrases in all official EU languages. Machine translation technology (currently available for around 18 language pairs) is used when rapid access to a large amount of information in different languages is needed, or when some officials would like to draft a document in a language other than their mother tongue. Machine translation systems are used also as a basis for an eventual translation of a document. The amount of correcting required varies according to the document type. Speech recognition technology is used as well (currently for only 9 EU languages) for dictating text directly in a natural, continuous way, achieving a high degree of accuracy and efficiency. The ergonomic an health benefits are also obvious, as adverse physical effects associated with intensive typing and mouse use are reduced. [Dir07c]

The Directorate General for Interpretation (DG Interpretation) is the European Commission's interpreting service and conference organizer, and provides interpreters for about 50 - 60 meetings per day in Brussels and elsewhere. The language arrangements for these meetings vary considerably — from consecutive interpreting between two languages, for which one interpreter is required, to simultaneous interpreting into and out of 23 or more languages, which requires at least 69 interpreters. At present, the Council of the Union accounts for around 46% of the interpreting services provided, followed by the Commission with around 40%. There are more than 500 staff interpreters, accompanied by 2700 accredited freelance interpreters. [Dir07a]

When working for the European Commission, translators or interpreters must have a university-level education, a perfect knowledge of the target language (usually their mother tongue), and a thorough knowledge of at least two other official languages.

[2] http://scic.cec.eu.int/europa/

In 2006[3], the European Parliament has spent approximately 300 million Euro, i.e. 30% of its budget, for the interpretation and translation of parliament speeches and EU documents. In total, an amount of approximately 1.1 billion Euros are spent per year for the translating and interpreting services within the European Union, which is around 1% of the total EU budget. [VS06]

2.3 Challenges in Human Interpretation

According to [AKESH00], researchers in the field of psychology, linguistics and interpretation, like Henderson [Hen82], Hendricks [Hen71] and Seleskovitch [Sel78], seem to agree that simultaneous interpretation is a highly demanding cognitive task involving a difficult psycholinguistic process. These processes require the interpreter to monitor, store and retrieve the input of the source language continuously in order to produce the oral rendition of this input into the target language. It is clear that this type of difficult linguistic and cognitive operation will force even professional interpreters to resort to a kind of groping for words, a kind of lexical or synthetic search strategy.

2.3.1 Fatigue and Stress

Fatigue and *stress* affecting the interpreter negatively, leading to a decrease in simultaneous interpretation quality. In a study of the fatigue factor and behavior under stress during extended interpretation turns by Moser-Mercer and her colleagues [MMKK98], professional interpreters were told to work until they could no longer provide acceptable quality. It was shown that: (1) during the first 20 minutes, the frequency of errors rose steadily; (2) the interpreters, however, appeared to be unaware of this decline in quality; (3) at 60 minutes, all subjects combined committed a total of 32.5 meaning errors; and (4) in the category of nonsense, the number of errors almost doubled after 30 minutes on the task. Following Moser-Mercer, it can be concluded “that shorter turns do indeed preserve a high level of quality, but that interpreters cannot necessarily be trusted to make the right decision with regard to optimum time on performing this task (interpreting)”.

Besides extended interpretation turns, other factors influence the interpretation quality. In a study by McIlvaine Parsons [Par78], factors rated by interpreters as stressful are: speakers talking very fast, the lack of clarity or coherence by the speaker, the need for intense concentration e.g. in TV-shows, the inexperience with the subject matter, a speaker’s accent, long speaker utterances between pauses, background noise, and poor positioning of the speaker’s microphone relative to the speaker. The stress factor was

[3] Until that time, only 20 official languages were available.

also compared between experts and novices in [Kur03]. She came to the conclusion that “conference interpreters have learned to overcome their stage fright with experience and have developed more tolerance for the stress involved in simultaneous interpretation, while student interpreters still grapple with numerous problems”.

In [Vid97], the conclusion was drawn that interpreters should work in teams of two or more and be exchanged every 30 minutes. Otherwise, the accuracy and completeness of simultaneous interpreters decrease precipitously, falling off by about 10% every 5 minutes after holding a satisfactory plateau for half an hour.

2.3.2 Compensatory Strategies

In experiments with students and professional interpreters Al-Khanji [AKESH00] found that the most frequent *compensatory strategies* are — in the order of occurrences — *skipping*, *approximation*, *filtering*, *comprehension omission*, and *substitution*. In order to get a deeper insight to the challenges of simultaneous interpretation for humans the strategies found during the experiments in [AKESH00] are summarized shortly.

Skipping: This strategy was used when: (1) the input is incomprehensible for the interpreter; (2) the interpreter decided that the input is repetitive; or (3) the interpreter was lagging behind the speaker.

Approximation: When there was no time for details, the interpreters attempted to reconstruct the optimal meaning by giving a less precise meaning of a word or an expression in the target language instead of the required lexical expression in the source language. Since enough semantic components were given in most cases, the meaning of the intended message was not negatively influenced.

Filtering: This strategy was used when the interpreter tried to compress the length of an utterance in order to find an economic expression. In so doing, interpreters seemed to preserve the semantic content of the message. Filtering is different from skipping in that interpreters are not necessarily facing a problem with the difficulty of economizing by reducing the length of an utterance.

Incomplete Sentences: Unlike skipping, the provision of incomplete sentences was used when interpreters omit larger units of speech, which may have resulted from a failure in text comprehension. In such cases, the interpreter initially made an attempt to start interpreting units of speech, which caused comprehension problems, but then gave up and cut short by stopping in mid-sentence.

Substitutions: This strategy was employed when interpreters used a lexical item in the target language which did not communicate the desired concept nor did it basically retain the meaning of the item in the source language.

2.3.3 Fluency and the Ear-Voice-Span

Since a audience is only able to evaluate the simultaneously interpreted discourse by its form, the *fluency* of an interpretation is of utmost importance. According to a study by Kopczynski [Kop94], *fluency* and *style* was third on a list of priorities of elements rated by speakers and attendees that contribute to quality, after content and terminology. Following the overview in [Yag00], an interpretation should be as natural and as authentic as possible, which means that artificial pauses in the middle of a sentence, hesitations, and false-starts should be avoided [Jon98] and the tempo and intensity of the speaker's voice should be imitated [Kop94].

Another point to mention is the time span between a source language chunk and its target language chunk, which is often referred to as *ear-voice-span*, delay, or lag. Following the summary in [Yag00], the ear-voice-span is variable in duration depending on some source and target language attributes, such as speech delivery rate, information density, redundancy, word order, syntactic characteristics, etc. Nevertheless, the average ear-voice-span for certain language combinations has been measured by many researchers, and varies largely from two to six seconds [Bar69, Led78], depending on the speaking rate. Short delays are usually preferred for several reasons. The audience is for example irritated when the delay is too large and is soon asking whether there is a problem with the interpretation. Another reason is that a short delay facilitates the indirect communication between the audience and the speaker but also between people listening to the interpreter and to the speaker. Therefore, interpreters tend to increase their speaking rate when the speaker has finished.

2.3.4 Techniques for Simultaneous Interpretation

Seleskovitch [Sel78], a professional interpreter and instructor for interpreters, advocates retaining the meaning of the source language utterance, rather than the lexical items, and argues that concepts (semantic storage) are far easier to remember than words (lexical storage). Semantic storage also allows the interpreter to tap into concepts already stored in the brain, which allows the interpreter to hitch a “free ride” on the brain's natural language-generation ability, by which humans convert concepts to words seemingly automatically. For this reason, preparation before a conference, by talking to the speaker and by researching the domain of the talk, is vital for interpreters. But Seleskovitch admits that concept-based interpretation may not always be possible. If the interpreter is unable to understand the concept being translated, or is under particular stress, they may resort to word-for-word translation. According to Moser-Mercer [MM96], also a professional interpreter and teacher and active in interpretation research, simultaneous interpretation must be as automatic as possible — there is little time for active thinking processes. The question is not avoiding mistakes — it is rather correcting them and moving on when they are made. Another suggestion for interpreters from Hönig [Hö97] is that interpreters, who must keep speaking in the face of incomplete sentences, must either "tread water" (stall while waiting for more input) or "take a dive" (predict the direction of the sentence and begin translating it). He suggests that "diving" is not as risky as it sounds, provided the interpreter has talked with the speaker beforehand, and has what he calls a "text map" of where the talk is headed. [Loe98]

Chapter 3

Automatic Simultaneous Translation

A speech translation system consists of two major components: speech recognition and machine translation. Words in the recorded speech input are recognized and the resulting hypothesis is transmitted to the machine translation component, which outputs the translation. While this sounds relatively easy, especially for simultaneous translation which require real-time and low latency processing with good translation quality, several problems have to be solved. Furthermore, automatic speech recognition and machine translation, which have evolved independently from each other for many years, have to be brought together.

Recognizing speech in a stream of audio data is usually done utterance per utterance, where the utterance boundaries have to be determined with the help of an audio segmenter before they can be recognized. Especially when the audio data contains noise artifacts or even cross-talk[1], this strategy can be extremely useful, because such phenomena can be removed in advance, leading to an improvement in ASR performance. However, the techniques used in such audio segmenters often require global optimization over the whole audio data and are therefore infeasible for a simultaneous translation system. On the other hand, even a simple speech/ non-speech based audio segmenter will introduce additional latency, since the classification of speech/ non-speech frames has to be followed by a smoothing process to remove mis-classifications.

Almost all machine translation systems currently available were developed in the context of text translation and have to cope with differences between a source and target language such as different amount and usage of word ordering, morphology, composita, idioms, and writing style, but also vocabulary coverage. Only recently has spoken language translation

[1] With cross-talk, speech from others in the background, which is recorded by the speaker's microphone is defined.

attracted wider interest. So, in addition to the differences between a source and target language, spoken language differs from written text in style. While text can be expected to be mostly grammatically correct, spoken language and especially spontaneous or sloppy speech contains many ungrammaticalities, including hesitations, interruptions, and repetitions. In addition, the choice of words and the amount of vocabulary used differ between text and speech. Another difference is that utterances are demarcated in written text, using punctuation, but such demarcation is not directly available in speech. This is a problem, because traditionally almost all machine translation systems are trained on aligned bilingual sentences, preferably with punctuation, and therefore are expecting sentences as input utterances in the same style. But when a low latency speech translation system is required, sentences are not an appropriate unit, because especially in spontaneous speech they tend to be very long — up to 20-30 words. To cope with this problem, a third component is introduced, which tries to reduce the latency by resegmenting the ASR hypotheses into smaller chunks without a degregation in translation quality. Chapter 8 describes this component in more detail.

Figure 3.1 gives a schematic overview of the simultaneous translation architecture treated in this thesis together with required databases and models. From the continuous input stream of speech, the ASR component is producing a continuous stream of partial first-best hypotheses, which are resegmented into appropriate chunks for the SMT component. The SMT component translates each of these source language chunks into the target language. By using multiple SMT components translation can be done in parallel into different target languages at the same time. For delivering the translation output, different technologies may be used among which the most prominent are either subtitles or speech synthesis. A more detailed description will be given in Chapter 9.

In the next section, some related research projects will be described. Compared to the previous chapter, the advantages of automatic simultaneous translation over human interpretation will be discussed in Section 3.2. The demands on such a system will be formulated in Section 3.3. Finally, in Section 3.4, an overview of some application scenarios in which such a system could be of use are described.

3.1 Related Research Projects

In the past, systems developed within research projects and consortia such as C-Star[2], Verbmobil[3], Nespole[4], Enthusiast, Digital Olympics, and Babylon

[2] Consortium for Speech Translation Advanced Research, http://www.c-star.org

[3] http://verbmobil.dfki.de

[4] Negotiating through Spoken Language in E-Commerce, http://nespole.itc.it

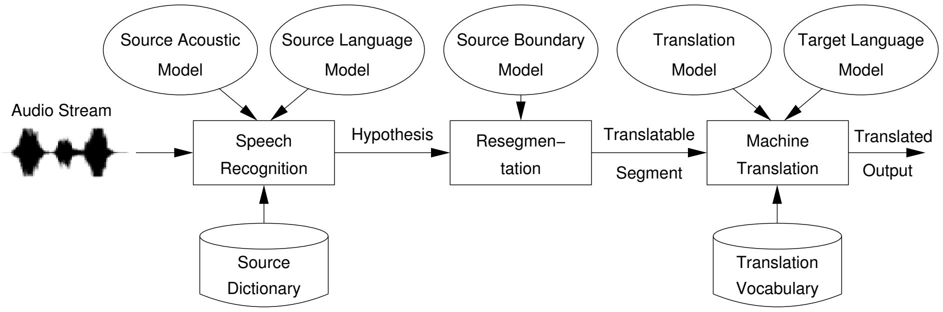


Figure 3.1: Schematic overview of the simultaneous translation architecture treated in this thesis. Boxes represent the components of the system, and ellipsis the models used by the components. As databases, dictionaries and vocabularies are used.

were able to support two-way communication, i.e. speech recognition and translation in limited application scenarios such as humanitarian aid, health care, tourism, government, etc., where the advantages of such a system outweigh the domain limitations. For these first systems, users had to speak in a well-behaved manner and the system was able to understand only a fixed number of phrase patterns. In almost all systems an *interlingua* [LGLW98] approach was used for translation and some of these systems were even able to run on handheld devices. Recently, in the DARPA-financed project Transtac[5], statistical-based machine translation systems were applied for use in a handheld device. Limited domain systems, available today, achieve a translation performance comparable to humans and are able to support human-human communication.

Nowadays, modern machine translation techniques can potentially provide affordable translation services to a wide audience, making it possible to overcome the language barrier for almost everyone. Thus speech-to-speech translation is attracting more and more attention. As a result of this, two major research projects were launched in Europe and USA focusing on open domain spoken language translation, TC-STAR and GALE.

TC-STAR — *Technologies and Corpora for Speech-to-Speech-Translation* [TS04], a European Commission-financed project, started in April 2004 and ended in April 2007 within the 6th Framework Program. It was envisaged as a long-term effort to advance research in all core technologies for speech-to-speech translation, including automatic speech recognition, spoken language translation and text-to-speech. The objective of the project was to make a breakthrough in speech-to-speech translation that significantly reduces the gap between human and machine translation performance. The focus was

[5] Spoken Language Communication and Translation System for Tactical Use, http://www.darpa.mil/ipto/programs/transtac/Transtac.asp

on the development of new algorithms and methods. The project targeted a selection of unconstrained conversational speech domains – speeches and broadcast news – and three languages: European English, European Spanish, and Mandarin Chinese. Project partners, mainly involved in speech recognition and/ or machine translation, were the Bruno Kessler Foundation (formerly ITC-IRST), the RWTH Aachen, LIMSI-CNRS, the Universitad Politècnica de Catalunya (UPC), Universität Karlsruhe (TH), and IBM.

The goal of the DARPA GALE – *Global Autonomous Language Exploitation* [GAL05] program is to develop and apply computer software technologies to absorb, analyze and interpret huge volumes of speech and text in multiple languages. Automatic processing engines will convert and distill the data, delivering pertinent, consolidated information in easy-to-understand forms to military personnel and monolingual English-speaking analysts in response to direct or implicit requests. In difference to TC-STAR, the output of each engine is English-translated text only and no speech synthesis is used. Instead, a distillation engine is responsible for integrating information of interest to its user from multiple sources and documents. The input to the transcription engine is speech, currently with a main focus on Arabic and Chinese. Military personnel will interact with the distillation engine via interfaces that could include various forms of human-machine dialog (not necessarily in natural language).

GALE evolved from two other past projects, EARS and TIDES. The goal of the EARS – *Effective, Affordable, Reusable Speech-to-Text* program was to "produce powerful new speech-to-text (automatic transcription) technology whose outputs are substantially richer and much more accurate than currently possible. The program focused on natural, unconstrained human-human speech from broadcasts and telephone conversations in a number of languages. The intent was to create core enabling technology suitable for a wide range of advanced applications, but not to develop those applications. Inputs and outputs will be in the same language." The TIDES – *Translingual Information Detection, Extraction and Summarization* program instead developed robust technology for translingual information processing. The goal was "to revolutionize the way that information is obtained from human language by enabling people to find and interpret needed information, quickly and effectively, regardless of language or medium." TIDES tasks included information detection, extraction, summarization and translation focusing mainly on English, Chinese and Arabic.

Another project to mention is CHIL – *Computers in the Human Interaction Loop*. CHIL aimed in making significant advances in the fields of speaker localization and tracking, speech activity detection and distant-talking automatic speech recognition. Therefore, in addition to near and far-field microphones, seminars were also recorded by calibrated video cameras. The long-term goal was the ability to recognize speech in a real reverberant environment, without any constraint on the number or distribution

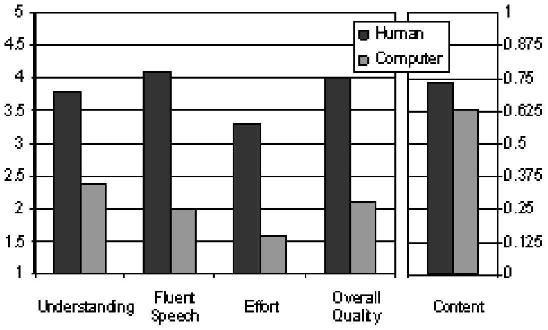


Figure 3.2: Comparison of automatic translation and human interpretation performance judged by humans in the project TC-STAR. [HMC07]

of microphones in the room nor on the number of sound sources active at the same time.

Parts of this thesis evolved within the two projects CHIL and TC-STAR.

3.2 Advantages of Automatic Simultaneous Translation

Given the explanations in the previous chapter of human interpretation in general and in the European Commission in particular, one has to weigh two factors when considering the use of simultaneous translation systems: cost and translation quality. The comparative results of TC-STAR in Figure 3.2 [HMC07] between human interpretation and automatic speech translation show that automatic translation was judged worse than human interpretation in most categories, but when it comes to the transfer of content both were judged nearly equally good. The reason why human interpretation does not reach "perfect" results is because often interpreters make use of the above mentioned compensatory strategies. On the other hand, even the automatic translation system can be of great help, especially for people not understanding the speaker's language at all. Furthermore, an automatic system can easily make use of additional information available about the speaker or the topic of the speech by using adaptation techniques to improve its perfomance. Note that the automatic TC-STAR system used for the comparison above was not working in real-time. This means that for a simultaneous translation system which has to deliver the translations with a latency as small as possible, a degregation in translation quality can be expected.

Another advantage of a simultaneous translation system compared to a human interpreter is that memorizing is not a problem for the system. Therefore the compensatory strategies *skipping*, *approximation*, or *incom-*

plete sentences described in Section 2.3 will not be needed, independently of the speaking rate of the speaker. However, depending on the system's translation speed it might be possible that the latency will rise. While it might be possible for humans to compress the length of an utterance without destroying the meaning, i.e. *filtering* or *summarization*, it is still a very challenging task for automatic systems [Fur07, Man01].

Another argument is that simultaneous interpretation at 300 to 400 Euros per hour, is quite expensive. The reason for that is that usually two interpreters are necessary and that the time for preparation and postprocessing must be considered additionally. Furthermore, simultaneous interpretation requires a soundproof booth with audio equipment, which can be rented, but this incurs additional costs which may be in most cases unsuitable for small events. On the other hand, a simultaneous translation system needs time and effort for preparation and adaptation towards the target application, language and domain. Depending on the required translation quality, the costs therefore can exceed those for a human interpretation. However, the major advantage of an automatic system is that once it is adapted, it can be easily re-used in the same domain, language etc. A single laptop together with a microphone is sufficient.

To some extent even generalization should not be a problem for automatic systems. Due to the way such systems are trained, expressions not directly within the required domain but closely related to it are already covered by the system. Furthermore, adaptation techniques can be used to extend the coverage and quality.

Especially in situations where a simultaneous translation into multiple languages is required, an automatic system is advantageous, because only the translation has to be extended to a new target language, while the source side recognition and resegmentation component of the system can be kept unchanged.

Another advantage is that the transcript of a speech or lecture is produced for free by using an automatic system in the source and target languages. In the European Union, for example, these transcripts can be used as an initial version of the protocols which have to be prepared anyway.

3.3 Demands on Automatic Simultaneous Translation

In comparison to other translation systems, and also given the observations regarding human interpretation in the previous chapter, four main demands on an automatic simultaneous translation system can be formulated:

- correct content
- correct syntax

- high fluency
- low latency

Obviously, a correct content is the most important demand on a simultaneous translation system. In connection to this, syntax also plays an important role, because the wrong syntax can destroy the content of a sentence. Nevertheless, to understand the content, a translation need not be completely correct. Instead, it may be sufficient if the words carrying the content or meaning of a sentence are correctly translated and no misleading content do exist.

Fluency on the other hand, requires that a simultaneous translation system produce a translation which is as natural as possible. Both content and syntax contribute to naturalness to some extent, but hesitations, false-starts, and other disfluencies, all characteristics of spontaneous speech, should also be removed either in advance or during translation [RLS07a]. Imitating the speaker's tempo also contributes to higher fluency, but also influences the latency. As already clarified in Section 2.3, it is important to keep the latency, i.e. the ear-voice-span of the whole system, as short as possible. In English, a latency of about two to six seconds is equivalent to a delay of about 4 to 12 words, since the average speaking rate is about 120 words per minute [RHP03, YLC06].

A comparable simultaneous translation system should therefore be able to produce speech translations of sufficient quality in real-time with a low latency.

3.4 Application Scenarios – The Lecture Scenario

Given the limitations of the recognition and translation capabilities of current speech translation systems, and the system development costs compared to human interpreters, possible application scenarios for simultaneous translation are restricted to domains to which a system can be well adapted and applications in which a system can be re-used and modified or customized with less effort. Therefore, the lecture scenario is selected as target scenario for this thesis, in which a single speaker is talking about a specific topic to a larger audience (see Figure 3.3). Small talks, student seminars or parliamentary speeches also belong to this scenario.

Other environments in which such a system could be of great use are telephone conversations and meetings. In both situations, it would allow people to communicate with each other independently of the language barriers. Telephone conversations and meetings are highly spontaneous dialogs and discussions between two or more people focusing on different topics and therefore difficult for both speech recognition and machine translation. Furthermore, a very low-latency simultaneous translation system is required

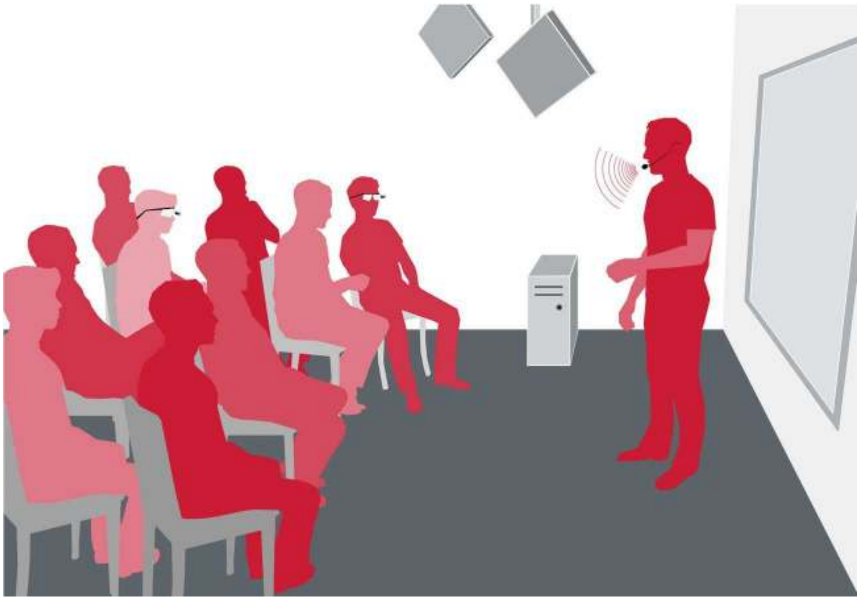


Figure 3.3: The lecture scenario. The speaker in front of the audience is recorded with the help of a microphone. The speech is transferred to the simultaneous translation system running on a PC or laptop and translated. The figure shows also different possibilities of how the translation output can be delivered to the audience: as subtitles on the projection screen, by using loudspeakers, or projected into heads-up display goggles.

because otherwise direct communication between the participants will be hindered. Thus, automatic simultaneous translation in these environments will remain challenging for several years.

Chapter 4

A First Baseline System

This chapter introduces a first baseline system for spoken language translation of lectures and speeches. This system will be suitable for consecutive translation only; the necessary techniques for supporting simultaneous translation will be presented later in Chapters 5 – 8. Starting from speech recognition and machine translation systems developed and used successfully in the context of the NIST RT-06S Rich Transcription Meeting Evaluation on lecture meetings and of the 2007 TC-STAR Evaluation on European Parliamentary Speeches, it will be shown how a first baseline spoken language translation system was built. It will be analyzed how both, the lectures recorded within CHIL and the speeches recorded within TC-STAR compare to the lectures on which we focus in this thesis with respect to recognition and translation quality.

The NIST Rich Transcription Meeting Evaluation series focused on the rich transcription of human-to-human speech, i.e. speech-to-text and meta data extraction with the goal to develop recognition technologies that produce language-content representations (transcripts) which are understandable by humans and useful for downstream analysis processes. The evaluation in 2006 (RT-06S[1]) was supported by two European projects, AMI – Augmented Multi-party Interaction[2] and CHIL – Computers in the Human Interaction Loop [WSS04]. Therefore, two data tracks were available on which a system could be evaluated: Conference Meetings and Lecture Meetings. In both data tracks, the speakers were recorded with close and far-field microphones, i.e. table-top microphones and microphone arrays. Conference meetings (supported by AMI) are goal-oriented small conference meetings like group meetings and decision-making exercises involving 4-9 participants, who are usually sitting around a table. In contrast, lecture meetings (supported by CHIL) are educational events, where a single lecturer is briefing an audience on a particular topic.

[1] http://www.nist.gov/speech/tests/rt/2006-spring/

[2] http://www.amiproject.org/

The European Parliament Plenary Speech (EPPS) task within TC-STAR focuses on transcribing speeches in the European Parliament in English and Spanish, and translating and synthesizing them into the other language. For the speech-to-text task, the data has been recorded from the European Union's TV Information service Europe by Satellite (EbS) [3], which broadcasts the sessions of the European Parliament live using separate audio channels for the speaker as well as for the simultaneous interpretations into all official EU languages [GBK+05]. Thus, the audio recordings contain speech from the politicians at their seats and at the podium using stand microphones, and speech from the interpreters using head-sets. For spoken language translation, the final text editions from the European Parliament are available through the EuroParl web site [4].

Since lectures and speeches are usually given in rooms with a large audience, microphones with a small distance to the speaker's mouth, such as head-worn or directional stand microphones are preferred over far-field microphones which are more sensitive to background noise. Therefore, the speech recognition systems were developed and optimized with respect to close-talk recording conditions. Since this thesis evolved in the context of the European projects CHIL and TC-STAR, European accented English was of particular interest.

In Section 4.1 lectures and speeches are characterized by their spontaneity and difficulty for speech recognition and spoken language translation and compared to other speech data such as recordings of read speech, broadcast news and meetings. After describing the training, test and evaluation data used for the speech recognition and machine translation systems and experiments in Section 4.2, the development of a first baseline speech-to-speech translation system is shown. For this purpose, a first speech recognition system is presented in Section 4.3 and, second, a first statistical spoken language translation system is introduced in Section 4.4. The performance, i.e. the recognition and translation quality and speed is measured and compared.

4.1 Characterization of Lectures and Speeches

In comparison to speeches given in parliament plenary sessions such as those in the European Parliament, lectures or talks are generally more difficult for speech recognition. The reason for that is that although potentially practiced in advance, lectures or talks are usually given freely and are therefore more spontaneous, thus containing more disfluencies and ungrammaticalities. In contrast, the speeches given in parliament plenary sessions are well prepared and often read. Furthermore, while the speeches given in parliament plenary sessions must be understandable for all politicians, the

[3] Europe by Satellite, http://europa.eu.int/comm/ebs/index_en.html

[4] The European Parliament Online, http://www.europarl.europa.eu

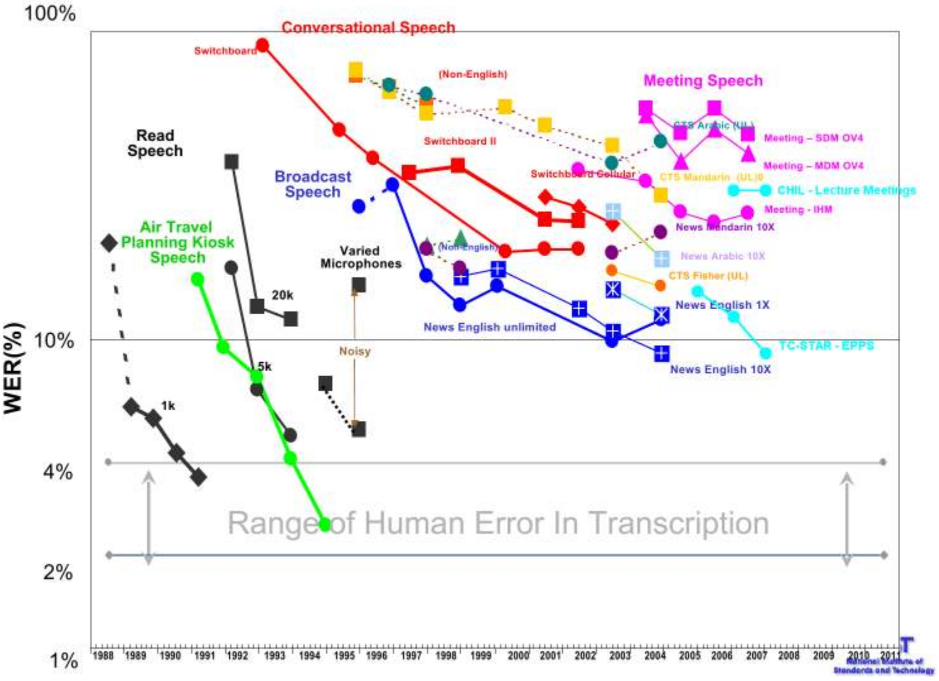


Figure 4.1: NIST benchmark comparison (from [FA07]). The speech recognition quality on tasks was measured in word error rate (WER).

level of detail in lectures or talks can be targeted to the audience, ranging from general understandable to very specific. E.g. the lectures recorded in CHIL are very technical oriented. On the other hand, when compared to meetings, both lectures and speeches are less spontaneous, more focused, and are mainly monologues with only small amounts of discussion, i.e. the question-and-answer part.

Figure 4.1 shows the evolution of word error rates over the years in speech recognition for several NIST-organized benchmarks. The results on European English lectures and parliament speeches were added from the projects CHIL and TC-STAR. From the figure, it can be observed that speech recognition quality decreases with an increasing amount of spontaneity in the speaking style and with domain complexity. The higher word error rates (see Section 4.2.5) for lecture meetings compared to meetings can be explained by the higher amount of research effort spent on meetings.

In an internal technical report [WB01], Burger analyzed the speaking style characteristics in the Translanguage English Database (TED), talks recorded at the Eurospeech Conference in 1993. Out of the corpus 51 speeches, between 5 to 22 minutes long, spoken by 48 individual speakers (37 male, 11 female) who originate from 14 different countries was used.

The speeches were categorized according to the English skill of the speaker and whether the speech was spontaneous or read. She observed that speaking rates decrease with English skill and are lower for read speech than for spontaneous presentations. Furthermore, the percentage of filled pauses, i.e. hesitations, and the percentage of corrections, i.e. fragments, repetitions, and false starts, were analyzed. The results showed that read speeches contained a lower frequency of filled pauses and corrections when compared to spontaneous ones, and that the amount of filled pauses and corrections increases with decreasing English skill. For average non-native English speakers, there were approximately 8% filled pauses and 6% corrections in the words spoken, while for native speakers the respective amounts were about 4% in both cases.

Unfortunately, there is no benchmark comparison available for (statistical) machine translation, since extensive benchmarking between different research sites for MT is relatively new, and differences in MT results are more due to the target language than the domain. However, a huge systematical analysis was done by Koehn for European languages on a corpus of European Parliament Plenary Sessions [Koe05]. He compared the translation scores for SMT systems for all language pairs of 11 different languages. Figure 4.2 shows some of the results for translation from and into English. It can be seen that for the language pair English-Spanish, which is used for the simultaneous translation system in this thesis as well, almost the best translation scores could be achieved. One reason for that is certainly that this language pair has been most heavily investigated by the research community.

4.2 Training, Test and Evaluation Data

In this section, the data used for training the acoustic, translation and language models of the speech recognition and machine translation systems is described. Furthermore, the details of the development data as well as the evaluation data used to tune the systems and for performing the experiments described in the following chapters are given.

4.2.1 Acoustic Model Training Data

As already mentioned in the introduction to this chapter, spontaneous, close-talk, European accented English is the focus of this thesis. Based on some experiments ([FIK$^+$06, FKB$^+$06]) for acoustic model training, recordings from meetings, European Parliament Speeches, lectures and student seminars were selected.

ICSI 72h of meetings recorded by the International Computer Science Institute (ICSI) with head-mounted and far-field microphones. The corpus

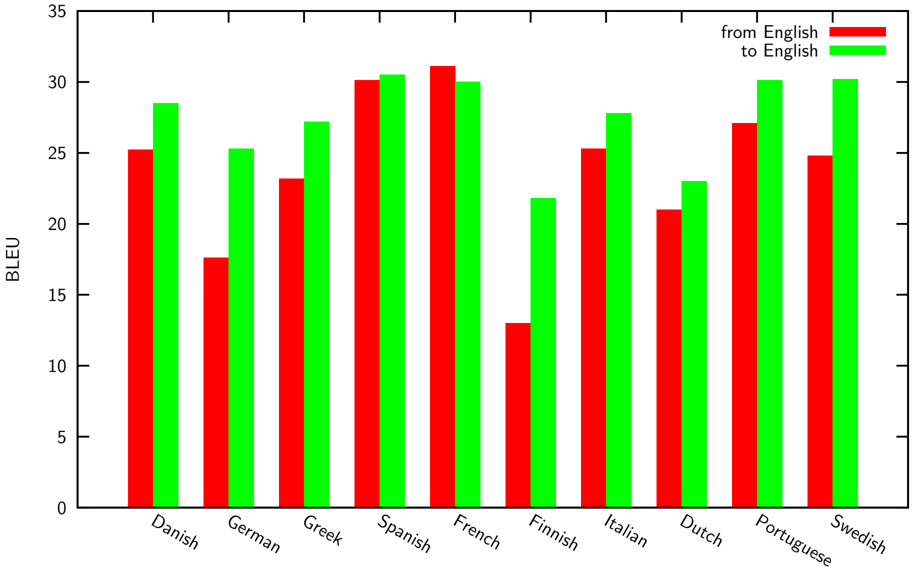


Figure 4.2: Machine translation results on European Parliament Plenary Sessions for translation between English and other European languages (from [Koe05]). Machine translation quality was measured automatically and are given in BLEU scores (see Section 4.2.5).

Type	Meetings		Speeches		Lectures	
	ICSI	NIST	EPPS	uEPPS	TED	Seminars
Duration [hrs]	72	15	80	167	10	10
Speakers	53	61	1894	2982	52	67
Recordings	75	19	63	74	39	17

Table 4.1: Summary of the data used for acoustic model training.

contains a significant portion of non-native English speakers, varying in fluency from nearly-native to challenging-to-transcribe [JAB+04]

NIST 13h of meetings recorded by the National Institute of Standards and Technology (NIST) with head-mounted and far-field microphones. This corpus contains mostly native American English speakers [GLM+04].

EPPS-S 80h of European parliament plenary sessions (EPPS) recorded within the TC-STAR project containing mostly non-native European English speakers [GBK+05].

EPPS-U 80h of European parliament plenary sessions (EPPS) recorded and transcribed automatically by RWTH Aachen within TC-STAR [GHSN07].

TED 10h of lectures recorded at Eurospeech 1993. The Translanguage English Database (TED) audio recordings have non-native English speakers presenting academic papers for approximately 15 minutes each [LSF+94, Cond].

SMNR 10h of mostly student seminars recorded within the CHIL project with head-mounted and far-field microphones, most of which were presented by non-native European English speakers [WSS04].

The most suitable data for lecture recognition are the lectures and seminars collected in the CHIL project (*SMNR*) and the Translanguage English Database (*TED*). Since the amount of data is insufficient to train acoustic models which perform well in large vocabulary speech recognition tasks such as lecture recognition, other data has to be used as well. The meeting data consists of recordings of highly spontaneous meeting speech from mostly native English speakers collected at three different sites: ICSI, NIST and CMU. In contrast, the European Parliament data covers non-native, usually well prepared speeches. An overview of the details of the several training corpora is given in Table 4.1.

	English	Spanish
Sentences	1,162,176	
Words	27.7M	28.9M
Vocabulary	93,157	130,473
Singletons	34,543	45,400

Table 4.2: Summary of the bilingual data used for training the machine translation system.

4.2.2 Translation Model Training Data

The baseline translation system was trained on a corpus of sentence-aligned, parallel final text editions from the European Parliament, available through its web site. The data was crawled, pre-processed and sentence-aligned automatically by RWTH Aachen [GBK+05]. The corpus statistics of the pre-processed EPPS training corpora are shown in Table 4.2.

4.2.3 Language Model Training Data

The language models for the baseline speech recognition systems were trained on the corpora used for the RT-06S and 2007 TC-STAR evaluation systems [FIK+06, SFH+06]. Three different types of data were used: speech transcripts, written text, and text data collected from the world wide web[5]. Altogether, the following English corpora were available:

EPPS-S 750k words of EPPS transcriptions in non-native English [GBK+05]

EPPS-T 33M words of EPPS final text editions [GBK+05]

EPPS-U 1.4M words of automatically transcribed speech data [GHSN07]

MTG 1.1M words of meeting transcriptions (ISL, ICSI, NIST, LDC) mostly in non-native English [Con04]

AMI 200k words of meeting transcriptions from the AMI project of mostly non-native English [CAB+05]

TED 98k words of transcriptions from the Translanguage English Database containing talks held at Eurospeech 1993 in non-native English [LSF+94]

SMNR 45k words of seminar transcriptions from the project CHIL in non-native English [WSS04]

[5] More details on the web-collection will be given in the Chapters 4 and 6

SWB 4M word of transcriptions from Switchboard [(LD]

BN 131M words of broadcast news written text data [Cona]

UN 42M words of the English part of the United Nations Parallel Text Corpus v1.0 [Cone]

HNSRD 48M words of text data consisting of U. K. parliament debates [Conc]

GWRD 167M words of text data extracts from the Gigaword corpus [Conb]

PROC 23M words out of recent speech and translation related conference proceedings from 2002 until 2005

UW-M 147M words of data collected from the web by the University of Washington related to ISL, ICSI, and NIST meetings

UKA-M 124M words of data collected from the web by ourselves related to *MTG* and *PROC*

UKA-L 146M words of data collected from the web by ourselves related to *SMNR*

UKA-LP 130M words of query-based filtered data collected from the web by ourselves related to *SMNR* and *PROC*

UKA-MP 124M words of query-based filtered data collected from the web by ourselves related to *MTG* and *PROC*

For the machine translation systems, the same parallel data for language model training was used as described above in Section 4.2.2.

4.2.4 Development and Evaluation Data

For development and evaluation, an amount of 19 lectures and talks on different topics of a single non-native speaker were collected in different environmental conditions. This data is further referred to as *lectures*. Table 4.3 gives an overview of the data.

While *lectDev* and *lectEval* were used for system development and evaluation, *lectOther* was used for adaptation experiments. IDs referring to talks start with a '*t*' and to lectures with an '*l*'. All lectures are given as part of a course related to the field of speech and language processing at the university. The talks instead were given for different reasons such as keynotes, press conferences, project reviews, or just an overview about the work done at our research lab. Therefore, the talks are more general and do not have the same level of detail as compared to the lectures. An exception

	Talk/ Lecture	Duration [min]	Words	Translation
lectDev	*l003*	77	12316	
	t012	44	7833	
	t032	71	11566	
	t035	32	5519	SPA
	t041	12	1979	SPA
	t042	7	1192	SPA
	$\sum$	243	40405	
lectEval	*l043*	44	7745	SPA
	t036	32	4824	SPA
	t037	16	2339	SPA
	t038	51	4673	SPA
	t044	29	4523	
	$\sum$	172	24104	
lectOther	*l005*	69	11049	
	t010	64	11169	
	t011	11	1931	
	t033	49	7507	
	t034	80	12539	
	t039	7	1128	SPA
	t040	8	1356	SPA
	t043	7	1072	SPA
	$\sum$	295	47751	

Table 4.3: Partitioning of the available lectures/ talks into development and evaluation data, as well as a comparison of the duration and number of word per lecture. Lectures for which a translation into Spanish was available are tagged with *SPA*.

constitutes the longer talks *t032*, *t033*, and *t034* which were part of a course about speech and language processing for which the lecturer was invited.

As can be seen in Table 4.3, human translations were available for a limited amount of talks and lectures only. In addition, the amount of translations per sentence were restricted to only one. Therefore, the machine translation was developed and evaluated on a subset of *lectDev*, *lectEval*, and *lectOther* only. To limit the necessary effort for the human evaluation this set was reduced even more to *t036*, *t037* and *t038* summing up to 92 minutes and 14,908 words. Due to the fact that *t036* and *t037* were given after another, the combination of both is often referred to as *t036+*.

Since there was a recording problem in *t010*, this talk was not used for acoustic model adaptation.

In addition, some speech recognition experiments were performed on the official development and evaluation sets of the TC-STAR-07 EPPS evaluation, further referred to as *parliamentary speeches*, and the lecture meeting data track of the NIST RT-06S evaluation, further referred to as *lecture meetings*. Especially the lecture meetings are well suitable as an additional development set because of their similarity in speaking style and focus on a particular topic. However, the level of detail is even higher than the average in *lectDev* and *lectEval*, and many of the recorded speakers have a strong foreign accent. Compared to the talks and lectures introduced above, both characteristics are responsible for a more difficult recognition.

Moreover, prior to decoding all lecture meeting and EPPS data have to be automatically segmented. As a result of this, segment boundaries do not necessarily match with sentence breaks and depending on the segmentation algorithm and recordings, the segmentation algorithm might fail and classifies speech as non-speech or vice versa. Both might result in additional recognition errors. A special care has also to be taken with cross-talk. If this is classified as speech belonging to the current speaker, the recognized speech will count as errors. The question-and-answer part also included in the lecture meeting data makes this even more difficult. As will be seen later, this explains the much higher word error rates for the lecture meetings compared to the talks and lectures. For the details of the different segmentation approaches used, we refer to [LFS07, FIK+06, SFH+06]. The details of the different development and evaluation data are presented in Table 4.4.

4.2.5 Performance Measures

The overall performance of a simultaneous translation system always depends on the performance of its sub-components, i.e. speech recognition, resegmentation, machine translation, and speech synthesis. Furthermore, since the components are connected in series, errors introduced by one component are usually amplified by subsequent components.

	ID	Duration [min]	Words	Speakers	Recordings
Dev	RT-06Sdev	150	22258	24	67
	EPPSdev	192	28976	44	7
Eval	RT-06Seval	190	21243	70	70
	EPPSeval	180	30362	41	5

Table 4.4: Additional development data. RT-06Sdev is identical to the RT-05S evaluation data, EPPSdev is identical to the 2006 TC-STAR development set, and EPPSeval is identical to the 2007 TC-STAR evaluation set.

The system's performance can always be judged by humans. However, human evaluations are very time-consuming and costly, and, especially for technical oriented lectures, the results often depend on the background knowledge of the evaluators. As a result automatically computable performance measures are preferred especially during system development, to accelerate the turn-around-time of the necessary experiments. Furthermore, they provide a convenient way for comparing different systems.

In the following, some automatically computable performance measures will be described. They are used in this thesis to judge the quality and speed of the sub-components of the simultaneous translation system, as well as the system's overall latency. It should be noted that, besides the below-mentioned performance measures, also other measures exist in the literature.

PPL The *perplexity* is a measure of language model performance and is based on the *entropy*, a measure of uncertainty associated with a random variable in the field of information theory [Sha48]. Given a sequence of words $w_1, w_2, \ldots, w_m$ from a vocabulary $\mathcal{V}$, the entropy H is defined as

$$H = -\lim_{n\to\infty} \frac{1}{m} \sum_{w_1,\ldots,w_m} \left(P(w_1,\ldots,w_m) \log_2 P(w_1,\ldots,w_m)\right). \quad (4.1)$$

The summation over all possible word sequences can be discarded under the assumption that the source is ergodic and given a large enough value of m, H can be approximated by

$$\hat{H} = -\frac{1}{m} \log_2 P(w_1, w_2, \ldots, w_m). \quad (4.2)$$

Using this approximation, the perplexity of a language model can be defined as

$$PPL = 2^{\hat{H}} = \hat{P}(w_1, w_2, \ldots, w_m)^{\frac{1}{m}}, \quad (4.3)$$

where $\hat{P}(w_1, w_2, \ldots, w_m)$ is the probability estimate of the word sequence by the language model.

OOV-Rate Given a vocabulary $\mathcal{V}$ and a text corpus $\mathcal{C}$, the *OOV-Rate* is defined as the percentage of word occurrences in $\mathcal{C}$ which are not covered by the vocabulary $\mathcal{V}$

$$OOV = 100 * \frac{\sum\limits_{\forall w \in \mathcal{C}} \begin{cases} 1 & , w \notin \mathcal{V} \\ 0 & , \text{otherwise} \end{cases}}{\sum\limits_{\forall w \in \mathcal{C}} 1}. \tag{4.4}$$

WER The *word error rate* (WER) is the standard metric to measure the quality of a speech recognition system. It is derived from the Levenshtein or minimum edit distance and is defined as the minimum edit distance between a given hypothesis and its reference transcription normalized by the length N of the reference transcription in words. The minimum edit distance is defined as the minimum number of substitutions S, deletions D, and insertions I of words required to transform the hypothesis into the reference transcription.

$$WER = \frac{S + D + I}{N}. \tag{4.5}$$

BLEU The *BLEU score* [PRWZ02] together with the NIST score [Dod02] are the standard metrics for automatically measuring the quality of a machine translation system. Both are based on the idea of a modified n-gram precision based on n-gram co-occurrence statistics. BLEU is defined as the geometric mean of modified n-gram precision scores p_n multiplied by a brevity penalty. By rearranging the original published formula [Dod02], BLEU can be formulated as

$$BLEU = \exp\left(\sum_{n=1}^{N} w_n \log\left(p_n\right) - \max\left(\frac{L^*_{ref}}{L_{sys}} - 1, 0\right)\right), \tag{4.6}$$

where uniform weights $w_n = 1/N$ and n-grams up to length N are usually used. L^*_{ref} is the number of words in the reference translation that is closest in length to the translation being scored – which becomes important when the scoring is done against multiple references – and L_{sys} is the number of words in the translation being scored. The modified n-gram precision p_n is defined as

$$p_n = \frac{\sum\limits_i \begin{pmatrix} \text{the number of n-grams in segment } i\text{, in the} \\ \text{translation being evaluated, with a matching} \\ \text{reference co-occurrence in segment } i \end{pmatrix}}{\sum\limits_i \begin{pmatrix} \text{the number of n-grams in segment } i \\ \text{in the translation being evaluated} \end{pmatrix}}. \tag{4.7}$$

NIST Compared to the BLEU score, the *NIST score* [Dod02] mainly differs in two characteristics. First, it uses an information criterion via which n-grams that occur less frequently, and are therefore more informative, are weighted higher. Second, it differs in the brevity penalty factor insofar that the impact of small variations in the translation length is minimized. The formula for calculating the NIST score is

$$NIST = \sum_{n=1}^{N} \left(\frac{\sum\limits_{\substack{\text{all } w_1 \ldots w_n \\ \text{that co-occur}}} \log_2 \left(\frac{\#\text{ of occurrences of } w_1 \ldots w_{n-1}}{\#\text{ of occurrences of } w_1 \ldots w_n} \right)}{\sum\limits_{\substack{\text{all } w_1 \ldots w_n \\ \text{in sys output}}} (1)} \right) \cdot \tag{4.8}$$

$$\exp\left(\beta \log_2 \left(\min \left(\frac{L_{sys}}{\bar{L}_{ref}}, 1 \right) \right) \right),$$

where $N = 5$, and β is chosen to make the brevity penalty factor $= 0.5$ when the number of words in the system output is $2/3$ of the average number of words in the reference translation. $\bar{L}_{ref}$ is the average number of words in the reference translation, averaged over all translations, and L_{sys} is the number of words in the translation being scored.

RTF The *real-time factor* describes the ratio between the duration d of an input and the time p necessary to process that input:

$$RTF = \frac{d}{p}. \tag{4.9}$$

The real-time factor is always machine dependent and is always computed in this thesis on an Intel Pentium D with 3GHz and 4GB of memory running under SuSE Linux 10.0. The Janus executable was compiled with the Intel C++ Compiler v9.1 using auto-vectorization.

LAT The *latency* describes the delay of a system between the input at time i and the processed output of the given input at time o:

$$LAT = o - i. \tag{4.10}$$

In this thesis, the latency describes the delay until a given speech segment is recognized, translated, and output, and will be measured in words or seconds.

PRC, RCL, and F-Measure *Precision* (PRC) and *recall* (RCL) are two measures widely used to evaluate the quality of information retrieval systems. In this context, recall describes the completeness of a search result and precision its accuracy. In the context of speech recognition, precision and recall can be defined as

$$PRC = \frac{C}{M} = \frac{C}{C+S+I}, \quad (4.11)$$

$$RCL = \frac{C}{N} = \frac{C}{C+S+D}, \quad (4.12)$$

where C is the number of correctly recognized words, S the number of substitutions, D the number of deletions, and I the number of insertions. M and N correspond to the number of words in the hypothesis or reference, respectively.

In the interest of having a single performance measure, the *f-measure* is used, which is defined as the weighted harmonic mean of PRC and RCL:

$$F = \frac{PRC \cdot RCL}{(1-\alpha)PRC + \alpha RCL}, 0 \leq \alpha \leq 1. \quad (4.13)$$

Usually an $\alpha = 0.5$ is used [MKSW99].

It is known from the literature [KP02] that there is a positive almost linear correlation between PPL and WER. The WER also depends on the OOV-Rate. As a rule of thumb, it can be said that an occurrence of an OOV word is average responsible for $1.5 - 2$ errors [GFS98].

From results obtained within TC-STAR and from our studies in [SPK+07] it can be seen that there is an almost negative linear correlation between the WER and machine translation quality. In addition the resegmentation component between speech recognition and machine translation has an influence on the machine translation quality. This dependency will be analyzed in Chapter 8.

Although it was shown in [SDS+06] that automatic measures for machine translation quality correlate with human judgment, current automatic measures are only to some extend able to judge semantic differences. This means that errors produced by the MT as well as ASR which affect a few words only, but destroy the semantic meaning of the whole sentence, are underestimated by all automatically computable performance measures. Therefore, the end-to-end performance of the simultaneous translation system, will be also judged by humans in Chapter 10.

4.3 Speech Recognition

In this section, a first baseline speech recognition system for lectures and speeches is introduced. As a starting point the evaluation systems build for RT-06S and TC-STAR-07 were used. Therefore, in the Sections 4.3.2 to 4.3.6 the differences and similarities of the systems components are described and compared with respect for their usefulness for lecture recognition. Section 4.3.7 presents a first baseline system using the results of the preliminary sections.

From an acoustic model point of view, optimization was done with respect to the word error rate on the lecture meeting data track of RT-06S, the parliamentary speeches of the 2007 TC-STAR evaluation mentioned in Section 4.2.4, and the lecture development data used in this thesis. In contrast, the language models were tuned with respect to the specific domain. This development strategy allows for a direct comparison to existing evaluation systems within the research community.

Acoustic model training was done with the help of the Janus Recognition Toolkit (JRTk) [FGH+97], which was jointly developed at Carnegie Mellon University, Pittsburgh, USA and Universität Karlsruhe (TH), Germany. For decoding and lattice generation the single-pass Ibis decoder [SMFW01], also part of the JRTk, was used. The language models used for decoding and lattice rescoring were built using the SRI Language Modeling Toolkit [Sto02].

4.3.1 Related Work

Research in the domain of the automatic transcription of lectures and speeches has gained interest, especially in the last couple of years. To date, several research groups and projects across the world deal with this topic. Within the European Commission-financed project CHIL, international evaluations on lecture meetings were carried out in the context of the NIST Rich Transcription Evaluation 2006 [LBA+06, HWC+06, FIK+06]. Besides this, also other research has been conducted, e.g. on the TED corpus [LSF+94, CBF04]. The focus of the European-funded project TC-STAR was the transcription of European Parliament Plenary Sessions [RSM+06a, LBG+06, LGA+07, SFH+06].

LECTRA was a national Portuguese project focusing on the production of multimedia lecture content for e-learning applications, which implies taking the recorded audio-visual signal and adding the automatically produced speech transcription as captions [TNN06]. The goal of the MIT Spoken Lecture Processing Project [GHC+07] is to improve the access to on-line audio/visual recordings of academic lectures by developing tools for the processing, transcription, indexing, segmentation, summarization, retrieval and browsing of this media.

In the Liberated Learning Consortium, the goal is to provide real-time, high-quality automatic speech transcription to aid hearing-impaired students. Research is conducted not only on improving the recognition quality, but also on the delivery aspect [BBFK05]. Real-time processing is also required for [RS02], where a lecture and presentation tracker is presented.

The biggest research effort on a single spoken lecture corpus has been in Japan using the Corpus of Spontaneous Japanese, which consists of about

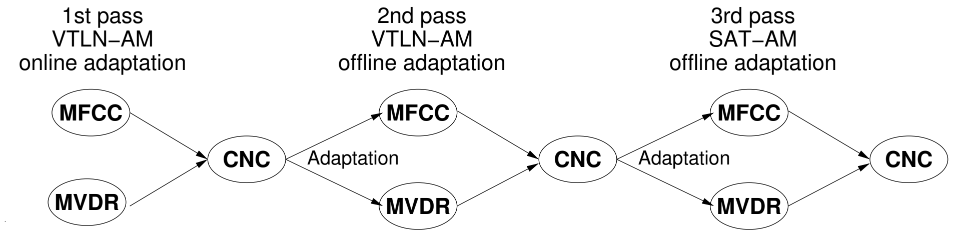


Figure 4.3: A typical multi-pass (here 3 passes) decoding strategy using two differently designed speech recognition systems (MFCC and MVDR). The final hypotheses are those of the last confusion network combination (CNC) pass.

700hrs of speech [Fur05].

4.3.2 Decoding Strategies

The appropriate decoding strategy plays an important role in a time-unlimited speech recognition system. To achieve the best recognition accuracy am offline system decodes a given utterance multiple times with different systems; whereas the acoustic models of one system are adapted in an unsupervised way on the output of a preliminary system. In addition, system combination techniques like ROVER [Fis97] or Confusion Network Combination [MBS99] are used to fuse the outputs of several different systems. An example of such a strategy used with minor modifications for the RT-06S and TC-STAR-07 evaluations is shown in Figure 4.3. The systems used, differ in the front-end (MFCC, MVDR) and in differently adapted acoustic models [FIK+06, FWM+06, SFH+06].

In contrast, simultaneous translation requires that all components run in real-time with low latency. Thus, for the following experiments, decoding was restricted to a single pass only, using one of the available front-ends and acoustic models.

Note that for the decoding results in this chapter incremental adaptation during decoding (online adaptation) is used, as it is the case for the first pass of the multi-pass strategy. Chapter 5 describes the type of adaptation used in more detail.

4.3.3 Front-End

In the front-end 13 Mel-scaled cepstral coefficients (MFCC) using a 16msec Hamming window with a frame shift of 10msec on 16kHz/16bit audio data are computed. After cepstral mean subtraction and variance normalization, which is updated per utterance taking speech frames only into account, seven adjacent frames to the left and to the right are stacked together to form a

195 dimensional feature vector. This vector is reduced to 42 dimensions using linear discriminant analysis (LDA).

4.3.4 Acoustic Models

The acoustic models for the two evaluations differ in the amount and type of training data, but the training procedure is more or less the same in both cases. For RT-06S, audio data from meetings (*ICSI*, *NIST*), lectures (*TED*) and seminars (*SMNR*) were used; for TC-STAR-07 parliamentary speeches (*EPPS-S*, *EPPS-U*) and lectures (*TED*) were used.

To simplify and speed-up the training procedure, fixed state alignments (labels) were used. In past experiments [SYM+04], it was shown that using a context-dependent system with a smaller acoustic model for label generation is advantageous than using a larger one. This was attributed to a better generalization capability of the small system. Moreover, systems trained according to these labels also outperformed those using Viterbi or forward/backward training.

The context-dependent models were created using an entropy-based clustering procedure. First mixture weights for all polyphone models were trained. This was followed by a top-down clustering procedure using a set of phonetic context- and position-dependent questions. Initially, the polyphone models rely on the context-independent models, but were consecutively split using a context of ±3 phonemes until 4000 context-dependent septa-phone models were reached. For the semi-continuous (SC) models, the decision tree of the fully continuous (FC) system was further split down until a maximum of 16000 mixture weights was reached, whereas the number of codebooks were kept fixed. A set of 87 questions was used, defined over a phoneme set with 45 phonemes and 7 noises (mono- and multi-syllabic fillers, breath, laughter, general human and non-human noise, and silence). Only the phonemes (not the noises) were modeled context dependently. For each model, a left-to-right 3-state HMM topology with self-loops was used. The exception to this was silence, where only the last state had a self-loop. Transition probabilities were globally set to 0.3 for all transitions and kept fixed throughout the training.

The training procedure from [SYM+04] was extended by adding a second step of incremental growing of Gaussians after the diagonalization of the feature space (semi-tied covariances). Although the amount of improvement in WER is inconsistent over different evaluations, it was observed that for RT-06S the extended training procedure outperforms the unextended variant. The overall training procedure therefore consists of the following steps:

1. Generation of fixed state alignments

2. Linear Discriminant Analysis

3. Sample Extraction
4. Incremental Growing of Gaussians
5. Semi-Tied Covariance Training [Gal98]
6. Sample Extraction
7. Incremental Growing of Gaussians
8. Viterbi Training
9. Discriminative Training using MMIE [Pov05]
10. Speaker Adaptive Training (SAT) [AMSM96]

Steps 2 to 7 all work on the same fixed state alignments produced in the first step and allows for a fast training procedure. Starting with step 8, Viterbi training is performed to compensate for the sometimes misaligned labels. The last step is optional and will be described on more detail in Chapter 5..

Lectures

In a first set of experiments, we analyzed the impact of different acoustic model training data on the WER of the different development sets: *RT-06Sdev*, *EPPSdev*, and *lectDev*. The results for these experiments are shown in Table 4.5. The language model for the experiments on *lectDev* was taken from the RT-06S system, because of the lower perplexity of 163 compared to 255 of the TC-STAR-07 language model on *lectDev*. From the results, it can be clearly seen that for *RT-06Sdev* the meeting data is of utmost importance and for *EPPSdev* the EPPS data. While for TC-STAR-07 adding the automatically transcribed EPPS data improves the WER by almost 0.5% absolutely it is unsuitable for *RT-06Sdev* and *lectDev*. Overall, relatively good results on all development sets could be achieved when using *SMNR, ICSI, NIST, TED,* and *EPPS-S* for acoustic model training.

4.3.5 Vocabulary and Dictionary

For RT-06S, the dictionary contained 58.7k pronunciation variants over a vocabulary of 51.7k. The vocabulary was derived by using the corpora *BN*, *SWB*, meetings (*MTG, AMI*), *TED* and *SMNR*. After applying individual word-frequency thresholds to the corpora, the resulting list was filtered with `ispell` [GWK02] to remove spelling errors, and extended with a few manually checked topic words from a set of topic bigrams. The OOV-rate on *RT-06Sdev* was 1.09%. [FIK+06, FWM+06]

	RT-06Sdev	EPPSdev	lectDev
Past Experiments			
SMNR, EPPS-S	40.3%	20.8%	
+ TED	38.7%	20.1%	
+ TED, ICSI, NIST	34.1%	20.6%	
Recent Experiments			
SMNR, TED, ICSI, NIST	31.9%		14.4%
+ EPPS-S	31.9%	14.5%	13.6%
TED, EPPS-S	39.3%	14.2%	15.6%
+ EPPS-U	39.9%	13.4%	15.7%

Table 4.5: The impact of different acoustic model training data on WER. The "past experiments" were done using a less advanced acoustic and language model.

For TC-STAR-07, a British English vocabulary was built using all words from the EPPS transcripts (*EPPS-S*) and all words with more than three occurrences from the EPPS final text editions (*EPPS-T*). This led to a vocabulary of 40k words and a case-sensitive OOV-Rate of 0.60% on *EPPSdev*, where hyphenated words were split into their constituent pairs. The pronunciation dictionary had a size of 46.1k. The mapping from American to British English spelling was done with the help of `respell` [Avi02].

In both cases, pronunciations were either derived from already existing dictionaries, from the CMU dictionary v0.6 [CMU] or automatically generated using Festival [BT97]. Although the automatic generation of pronunciations using Festival is sometimes inconsistent and not perfect, e.g. especially proper names and acronyms are often pronounced incorrectly and pronunciation variants cannot be generated, a small experiment showed that the degradation in WER is not that large. A new system was trained using pronunciations from Festival only and the results were compared to an identically trained system using a pre-existing dictionary. The absolute difference in WER was only 0.6%.

Lectures

The comparative results in Table 4.6 show that the OOV-Rate for the RT-06S vocabulary on *lectDev* of 0.47 is much lower than that obtained with the TC-STAR-07 vocabulary. It should be noted that in advance to the OOV-Rate computation, the spelling of the vocabularies was normalized to either American English for RT-06S or British English for TC-STAR-07, using the afore-mentioned `respell`. Therefore the OOV-Rates differ slightly from the numbers mentioned in the system descriptions above or in [FIK+06, FWM+06, SFH+06].

	RT-06S		TC-STAR-07	
	OOV	PPL	OOV	PPL
RT-06Sdev	1.09	153	2.42	283
EPPSdev	1.35	276	0.60	84
lectDev	0.47	163	1.94	255

Table 4.6: In- and across-domain perplexity and OOV-Rate comparison of the vocabularies used for the RT-06S or TC-STAR-07 evaluation on different development sets (*RT-06Sdev*, *EPPSdev*, *lectDev*).

4.3.6 Language Model

For both evaluations a 4-gram mixture language model was used, with components trained on different corpora using Chen and Goodman's modified Kneser-Ney discounting and also an interpolation of the discounted n-gram probability estimates with lower-order estimates [CG98]. Pruning was performed after the interpolation of the language model components, using a fixed threshold of 10^{-9}. The language models will be only described briefly, a more detailed discussion is given in Chapter 6.

For RT-06S, the language model components were trained on the following corpora: meeting data transcripts (*MTG, AMI*), transcripts of lectures and seminars (*SMNR, TED*), text data (*BN, PROC*), and several corpora collected from the web (*UW-M, UKA-LP, UKA-MP*). The mixture weights for each language model were optimized on a held out set with a size of 30k words. The final language model is further referred to as *LM6*.

For TC-STAR-07 separate 4-gram language models were trained on each of the following corpora: the EPPS corpora (*EPPS-S, EPPS-T, EPPS-U*) and the text corpora (*BN, UN, HNSRD, GWRD*). The final language model was the result of interpolation of all separate 4-gram language models, with the interpolation weights tuned on the 2006 EPPS evaluation data by minimizing perplexity.

Lectures

As can be seen in Table 4.6, the perplexity on the lecture data of the language model for RT-06S (163) is much lower than the perplexity of the language model for TC-STAR-07 (255). The reason is the technical nature of the RT-06S, data which is more similar to the lectures used in this thesis.

4.3.7 Baseline System

In summary, the baseline system was trained on almost 190 hours of speech data from seminars, lectures, meetings and European Parliament plenary sessions, leading to a fully continuous acoustic model with approximately

		RT-06S		TC-STAR-07		lectures	
		dev	eval	dev	eval	dev	eval
SC	WER	30.4%		16.6%		15.5%	
	RTF	1.47		1.27		1.43	
FC	WER	29.3%		16.2%		15.1%	
	RTF	1.34		1.16		1.14	
FC-MMIE	WER	27.9%	35.6%	15.8%	14.3%	14.3%	15.7%
	RTF	1.28	1.46	1.12	1.20	1.04	1.21

Table 4.7: Word error rates (WERs), after language model rescoring, and real-time factors of the first baseline system on the different development and evaluation sets. *SC* stands for a semi-continuous and *FC* for a fully continuous acoustic model.

234k Gaussians divided amongst 4000 context dependent models in a 42 dimensional feature space. As described above, the features were extracted from the audio signal using MFCCs followed by a stacking of 15 consecutive frames and an LDA.

Table 4.7 compares the WERs and RTFs for different acoustic models on the development and evaluation sets. The rows marked with *SC* show the WERs and RTFs when using semi-continuous acoustic models. The semi continuous acoustic model was trained by splitting the 4000 context-dependent distributions further down to 16000 and leaving the number of codebooks fixed. The rows marked with *FC* show the WERs and RTFs for the systems based on fully continuous acoustic models. *MMIE* refers to an acoustic model which was trained using four additional iterations of discriminative training.

As already mentioned, for RT-06S and TC-STAR-07, the audio data was automatically segmented into speech and non-speech regions and clustered in homogenous speaker intervals. For the *lectures*, manual segmentation and true speaker labels were used. For the results on TC-STAR-07, the British English language model and vocabulary were used; for decoding the RT-06S and *lecture* data, the American English language model and vocabulary were used (both described in Section 4.3.6). During decoding, incremental adaptation as mentioned above was employed.

Since simultaneous lecture translation requires a system running in real-time, system design decision have to be made early. Therefore, compared to the RT-06S and TC-STAR-07 evaluation systems, decoding was speed-up by tightening the search beams so that the systems marked with *SC* and *FC* in Table 4.7 are running at comparable speed. With this constraint, it can be observed, that a fully continuous system outperform a semi-continuous. The WER drops for all development and evaluation sets. Furthermore, it can be seen that using a more sophisticated acoustic model leads to lower

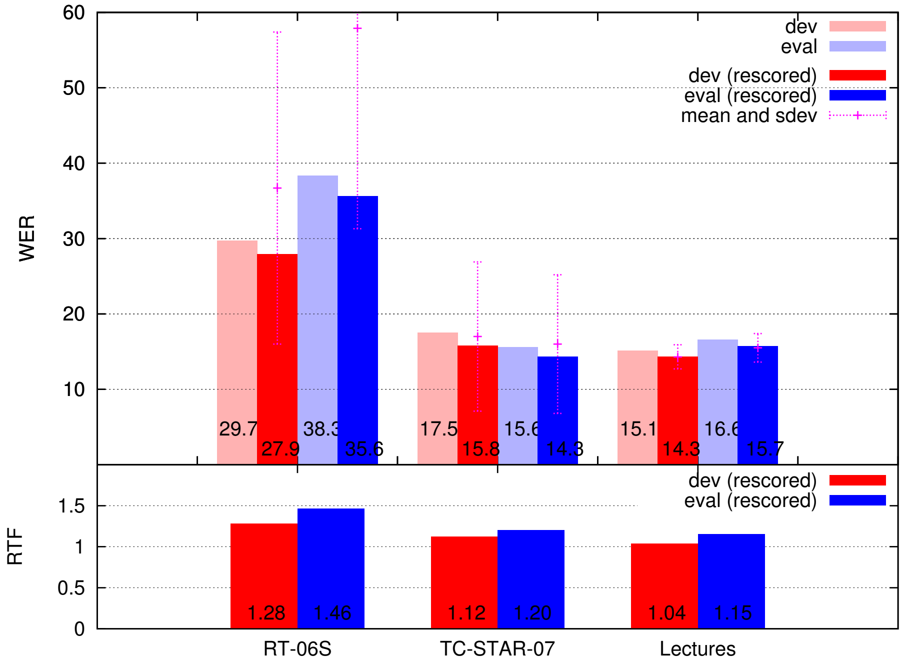


Figure 4.4: Comparison of WERs for the *FC-MMIE* acoustic model on the different development and evaluation sets before and after language model rescoring. Also shown are RTFs computed with language model rescoring. Lines within the WER boxes mark the mean WER and its standard deviation (sdev) per speaker in a lecture meeting, parliamentary speech, or lecture.

WERs and, because of this, also to lower RTFs. In this case the same beam settings were used as for the *FC* system. Using discriminative training improves WERs on all conditions by approximately 1% absolute, leading to a relative average speed-up of 5.8%.

In Figure 4.4, the WERs for the *FC-MMIE* acoustic model are compared and set in relation to the corresponding real-time factors. Furthermore, the impact of language model rescoring on the WER is analyzed, and the mean WER and its standard deviation per speaker in a lecture meeting, parliamentary speech, or lecture are shown. It can be seen that while parliamentary speeches and lectures can be recognized with an almost equal recognition accuracy and speed, the lecture meetings are much more difficult. Since the lectures were collected only from a single speaker, the variance in terms of WERs are very low compared to the variance for the parliamentary speeches and lecture meetings. Especially for TC-STAR-07, a significant number of speakers with WERs lower than those obtained on the *lectures* exists. It should be noted that for the *lectures* the mean WER and its standard deviation was computed across different lectures of the same speaker. For

RT-06S and TC-STAR-07, this was done across different speakers and lecture meetings or parliamentary speeches.

4.4 Statistical Machine Translation

In this section, we discuss the approach for developing the statistical machine translation component in our lecture translator that was used to translate lectures and speeches from English to Spanish. The initial purpose of the underlying phrase-based SMT system developed within TC-STAR was to translate speeches from the European Parliament Plenary Sessions in an offline scenario (that is, translating without constraints with respect to some parameters which are critical in a simultaneous translation scenario, such as processing time, resource usage, and latency). As described earlier, parliamentary speeches, while covering a relatively broad discourse domain, are usually prepared speeches given by well-trained speakers. Lectures, on the other hand, can go into much more detail on any given topic, and, as an aggregate, cover a practically unlimited domain; a system suitable for translating general lectures must be able to cope with much more variable and more spontaneous speaking styles. At the same time, the machine translation component can make use of the additional information about the topic, which is known in advance in a typical lecture translation scenario, and offer acceptable speed and latency in real-time operation. We therefore needed to develop a considerably different system from the one built and optimized for traditional batch-mode operation and well-controlled speech. In our experiments, we used loose coupling, passing the first-best hypothesis from the recognizer to the translation component. All MT scores were calculated using case-insensitive scoring and one set of reference translations per test set.

4.4.1 Phrase Alignment

Phrase-to-phrase translations are critical for the performance of state-of-the-art statistical machine translation systems, and our lecture translation system builds upon this foundation. Methods used for the extraction of phrase translation candidate pairs from bilingual corpora are generally run during a training phase, prior to the decoding of unseen test sentences. For real-world tasks such as EPPS, broadcast news or lecture translation, these methods produce huge, multi-Gigabyte phrase translation tables containing hundreds of millions of translation alternatives, each with multiple feature annotations used for scoring and selection. Such raw phrase tables cannot normally be used directly for decoding.

In batch translation processing, the set of test sentences is small and known beforehand, so these phrase tables can be pruned to the specific test set in question. In a simultaneous translation scenario, however, this is not feasible if the system is to be able to translate on demand any conceivable input in a large or even unlimited domain.

Therefore, Vogel developed a novel approach to phrase alignment and extraction (PESA) which is particularly well-suited for efficiently providing phrase translation alternatives on-the-fly [Vog05]. This method, based on optimizing a constrained word-to-word alignment for an entire sentence pair, can be used to extract phrase translation candidates of arbitrary length from the training corpus at decoding time [KZV$^+$06].

For the experiments reported in this paper, we used a classic phrase table constructed by training IBM Model-4 word alignments in both directions, and extracting phrase-to-phrase translation pairs which are consistent with these word alignments. This significantly improve translation quality. The GIZA++ toolkit [ON03] and the implementation of the grow-diag-final heuristic provided by the University of Edinburgh [KM06] were used for training word and phrase alignments, respectively. In addition, we applied modified Kneser-Ney discounting to the raw relative frequency estimates of the phrases as described by [FKJ06]. Finally, the phrase table was pruned to the top 10 phrase translations for each source phrase using the combined translation model score as determined by Minimum Error Rate (MER) optimization on a development set.

4.4.2 Decoder

The beam search decoder combines all model scores to find the best translation. In these experiments, the different models used were:

1. The translation model, i.e. the word-to-word and phrase-to-phrase translations extracted from the bilingual corpus according to the PESA alignment method.

2. A trigram language model. The SRI language model toolkit [Sto02] was used to train the models.

3. A word reordering model, which assigns higher costs to longer distance reordering. [VNT96]

4. Simple word and phrase count models. The former is used to compensate for the tendency of the language model to prefer shorter translations, while the latter can be used to give preference to longer phrases. For each model, a scaling factor can be used to modify the contribution of this model to the overall score.

The decoding process is organized into two stages: first, the word-to-word and phrase-to-phrase translations (and, if available, other specific information such as named entity translation tables) are inserted into a translation lattice. In the second step, we find the best combinations of these

	NIST	BLEU
t036+, text input	5.72	23.4
t036+, ASR input	5.06	17.9
l043, text input	5.27	19.6
l043, ASR input	4.80	16.6

Table 4.8: Machine translation results on text as well as on ASR input on the talks/ lectures *t036+* and *l043*.

partial translations such that every word in the source sentence is covered exactly once. This amounts to performing a best path search through an extended translation lattice to allow for word reordering. Decoding proceeds essentially along the source sentence. At each step, however, the next word or phrase to be translated may be selected from all words lying or phrases starting within a given look-ahead window from the current position [Vog03]. The use of a local reordering window in the lecture translation system captures most of the benefits of improved target word order while keeping search complexity low for real-time operation.

4.4.3 Baseline System

For training the baseline translation system, the parallel EPPS corpus described in Section 4.2.2 was used. Although speech recognition is unable to deliver punctuation marks with its hypotheses, they were left in the corpus but separated from the words. The reason for this is that they help in finding useful split points for phrase alignment training. Misalignments in the corpus and other "noises" such as document references were discarded. Furthermore, abbreviations were expanded and dates and number expressions spelled out. In conformity with speech recognition, all data was lower-cased.

All punctuation marks were later removed from the source side of the phrase alignments to keep the interface between ASR and MT as simple as possible. As explained in Chapter 8, developing an additional punctuation annotation module with reasonable performance is quite challenging, and according to [MMN06], the performance of SMT is affected only slightly.

For training the initial language model, the target side (i.e. Spanish) of the same corpus was used. Modified Kneser-Ney discounting and interpolation of discounted n-gram probability estimates were applied. This resulted in a 3-gram language model covering 131k words, with a perplexity on the Spanish side of *lectDev* of 631.

Table 4.8 summarizes the translation results of the baseline system on the development and evaluation data for the *lectures* and TC-STAR-07. As expected, performance on ASR output is worse than on manually transcripts.

4.5 Conclusion

In this chapter, the speech recognition and machine translation components of the baseline system were presented. Starting from existing evaluation systems for RT-06S and TC-STAR-07, it was shown, how these systems can be combined to form a system suitable for lecture recognition and translation. Currently, both components operate independently. The next step is to focus on improving the components with respect to quality, speed, and latency. Overall, system recognition and translation quality is addressed with the help of speaker and topic adaptation techniques (Chapters 5 and 6), improving the overall system speed by limiting the size of the search space or number of parameters to be evaluated (Chapter 7), and limiting the system latency requires streamlining the interface between ASR and SMT (Chapter 7 and 8).

Chapter 5

Speaker Adaptation

In Chapter 4, a baseline speech translation system was developed. It was trained on a large amount of data, covering different speaking styles, speaker characteristics, spontaneous speech and topics in lectures and speeches recorded using different close talking microphones. However, such a general simultaneous translation system for lectures and speeches often does not work as well as a more specialized system.

On the other hand, lectures and speeches are particularly suited for adaptation, because usually the speaker and also the topic of a lecture is known in advance. This makes it possible to perform adaptation before the lecture starts either by using existing data or by collecting new data matching the speaker and/ or the topic. For example, one could use audio data of the same speaker for unsupervised acoustic model adaptation, select relevant text data out of existing data for language model adaptation, or crawl the World Wide Web for new matching data.

Adaptation can be performed on different levels or with respect to different models of a simultaneous translation system:

Acoustic Model Variations in the audio signal like ambient noise, different recording channel characteristics, or the speakers voice can be normalized at the signal level with respect to a specific acoustic model. On the other hand, also an acoustic model can be adapted to new input conditions.

Source Language Model Language model adaptation can be used to adapt a language model to a specific talk or lecture, but also to the speaker's speaking style, which can differ in the degree of spontaneity as well as in the selection of phrasing.

Translation Model Translation model adaptation can improve the translation quality in all situations where the training data does not match the new conditions, and is therefore used mainly for topic adaptation especially when new, unseen phrases or words have to be translated.

Target Language Model The target language model is used by the machine translation system and can be adapted for the same reasons and in the same way as the source language model.

Adaptation is not particular to automatic systems — human interpreters also adapt. Especially for simultaneous interpretation, human interpreters interview the speaker and familiarize themselves with the topic beforehand through e.g. literature research. Thereby, the translation of special phrases can be memorized in order to allow for a more fluent interpretation when required.

The remainder of this chapter will focus on speaker adaptation in the context of speech recognition only; topic adaptation will be treated in Chapter 6. First, some commonly used techniques suitable for speaker adaptation are described in detail in Sections 5.1 and 5.2. After this, the techniques described are applied to the current system either online during decoding in Section 5.3 or offline in advance to decoding in Section 5.4. The improvements due to both strategies separately and in combination is compared on the lecture data. Finally, in Section 5.5, the results of this chapter are summarized and concluded.

5.1 Adaptation Techniques

In general, adaptation techniques fall into two categories: normalization of the input to match the model and model adaptation techniques in which the parameters of the model are adjusted to better match the input. This section will describe some commonly used adaptation techniques, but concentrates on speaker adaptation in the context of a speech recognition system, wherefore the input is of a specific speaker only and the model is the acoustic model of the speech recognizer. An important issue with both approaches is how effective they would be with a limited amount of adaptation data relative to the size of the acoustic model.

A further distinction can be made between *supervised adaptation*, where the adaptation parameters are estimated with respect to reference transcriptions, and *unsupervised adaptation*, where system hypotheses are used instead. Some of the techniques can be used during acoustic model training only, others also during decoding for incremental adaptation.

5.1.1 Vocal Tract Length Normalization

Vocal tract length normalization (VTLN) is a feature transform which attempts to normalize the shift of the formant frequencies of different speakers caused by their different vocal tract lengths. The basis of this technique is the assumption that the positions of the formants in frequency are inversely proportional to the length of the vocal tract, and can therefore be scaled with the length of the vocal tract [Fan60]. In our case, a piece-wise linear function $f(w)$ is used to warp the frequency axis:

$$f(w) = \begin{cases} \alpha^{-1} w & \text{if } w \leq w_0 \\ \frac{1-\alpha^{-1}\beta}{1-\beta} w + \frac{\alpha^{-1}-1}{1-\beta} w_0 & \text{if } w > w_0, \end{cases} \tag{5.1}$$

where α is the speaker-specific warping factor, $w_0 = \beta \cdot w_s/2$ and w_s is the sampling frequency; the edge β is empirically set to 0.8. Speakers with longer vocal tracts, have warping factors of $\alpha \leq 1$, and for speakers with shorter ones an $\alpha > 1$ can be observed.

In [AKC94] a maximum likelihood approach was introduced to estimate the warping factors α of a specific speaker. Given a reference transcript or a recognizer hypothesis, the score of a Viterbi pass can be computed and compared for different α. Often a Brent search is utilized over an interval of warping factors to speed-up the search process. Since formants are observed for voiced phonemes only, for computing Viterbi scores only voiced frames are taken into account; however, the frequency warping afterwards is applied to all frames.

5.1.2 Maximum A-Posteriori Estimation

Given a model λ (parameter vector) and an observation x (input vector) the Maximum Likelihood method (ML) attempts to optimize the observation probability $p(x|\lambda)$. In difference thereto, the *Maximum A-Posteriori estimation* (MAP) [GL94] aims to optimize the parameter probability $p(\lambda|x)$, so that we get

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}}\, p(x|\lambda) \cdot p(\lambda) \tag{5.2}$$

with the priors $p(\lambda)$. MAP distinguishes itself from ML by only these priors.

In our case, we use a simplified count-based version of MAP, where the priors are defined over the relation of the number of speech training samples between each model multiplied with an additional weighting factor.

5.1.3 Maximum Likelihood Linear Regression

In the Maximum Likelihood Linear Regression (MLLR) framework, the Maximum Likelihood criterion is used to estimate a linear transform of the model parameters. This transform can be either applied to the models [LW95] or to the features [Gal97].

MLLR in the Model Space

In the context of mixtures of Gaussians, an adaptation of the means of Gaussians can be represented by probability density functions (PDFs) such as

$$p(o|s,\Psi) = \sum_i w_i N(o; A\mu_i, \Sigma_i) \tag{5.3}$$

with the observation vector o, the state s, and the acoustic model parameters $\Psi = \mu_i, \Sigma_i, w_i$ with the distribution weight w_i, the mean μ_i, and the covariance matrix Σ_i of a Gaussian mixture component i. Keeping the Gaussian parameters fixed, the Kullback Leibler statistics

$$Q(A, A^0) = c - \sum_{i,t} \gamma_i(t)(c_i + (o_i - A\mu_i)^T \Sigma_i^{-1}(o_t - A\mu_i)) \tag{5.4}$$

can be used to estimate the linear transform A. The state probabilities $\gamma_i(t)$ are computed using the initial parameter A^0. Terms not relevant for the optimization are denoted by c and c_i. The maximization of Q requires solving

$$\frac{d}{dA} Q(A, A^0) = 0 \tag{5.5}$$

Differentiating Q with respect to A leads to a set of linear equation systems, which can be solved row by row:

$$\sum_{i,j} \gamma_i(t)\Sigma_i^{-1} o_i \mu_i = \sum_{i,j} \gamma_i(t)\Sigma_i^{-1} A\mu_i\mu_i \tag{5.6}$$

MLLR in the Feature Space

Applying the linear transform in the feature space instead has some computational advantages over the model adaptation since combinations with adaptive training schemes and Gaussian selection algorithms are easy to realize. When transforming the features, it is not possible to transform means and covariances differently as is the case when transforming models, which is why this approach is also called “constrained MLLR” (cMLLR).

Given a PDF $p(x)$ and a feature transform $f(x)$, an appropriate PDF with respect to f would be $\hat{p}(x) = p(f(x))\frac{d}{dx}f(x)$. This ensures that the probability mass is conserved:

$$\int p(x)dx = \int p(y)dy = \int p(y)\frac{dy}{dx}dx = \int p(f(x))\frac{df(x)}{dx}dx = \int \hat{p}(x)dx \tag{5.7}$$

When $f : \vec{x} \rightarrow \vec{y}$ is a vector function, the corresponding substitution rule is extended to the functional determinant or Jacobian. The corresponding Kullback-Leibler statistics for a linear transform $f(x) = Ax$ are:

$$Q(A, A^0) = c + \sum_{i,j} \gamma_i(t)(\log|A| - c_i - \frac{1}{2}(Ao_t - \mu_i)^T \Sigma_i^{-1}(Ao_t - \mu_i)) \tag{5.8}$$

The Jacobian $|A|$ term complicates the optimization process. However, the Laplace development for a row j results in the following representation of the Jacobian:

$$|A| = \sum_{jk} a_{jk}\tilde{a}_{jk} \tag{5.9}$$

$$\tilde{a}_{jk} = (-1)^{j+k}|A_{jk}| \tag{5.10}$$

where $\tilde{a}_{jk}$ denotes the adjunct of A, given j and k. This allows for the implementation of an iterative row-by-row optimization scheme. The adjuncts $\tilde{a}_{jk}$ are kept fixed when optimizing row j.

5.2 Speaker Adaptive Training

Speaker adaptive training (SAT) [AMSM96] is a technique which is used to estimate the parameters of continuous density HMMs for speaker-independent speech recognition in a way that integrates speaker adaptation in the common speaker-independent training paradigm [MSJN97]. The differences amongst the speakers in the training data, which lie e.g. in the anatomy of the vocal tract, the speaking style, or accent of the speaker, may result in a diffused acoustic model with reduced discriminative capabilities. The reason for this is a higher variance in the spectral distribution as compared to a speaker-dependently trained system. Cepstral mean and variance normalization [AKMS94] is the simplest feature space-based normalization method to counteract channel effects; the above mentioned vocal tract length normalization is another one. In general, feature space-based normalizations, which are often based on linear transformations, are preferred over model space-based normalizations because they can be applied more easily and are usually computationally less expensive. For example the above-described constrained MLLR is the most commonly used technique for speaker adaptive training and can also be used together with other normalization techniques, either in the feature or model space. When applying more than one normalization technique, the question arises, in which order the normalization parameters have to be estimated to achieve the best performance.

A speaker adaptively trained system requires that the same speaker normalization techniques are applied during decoding. Without such normalization, systems will perform worse than compared to a speaker-independent system. The following section analyzes this problem in more detail.

5.3 Online Adaptation

With online adaptation, the application of adaptation during decoding is referred. Because of their simplicity, cepstral mean and variance normalization, vocal tract length normalization, and constrained MLLR are particularly well for online speaker and channel adaptation. For the following experiments, the adaptation is performed in the same way as for the first pass decoding of the evaluation systems for RT-06S and TC-STAR-2007. Briefly,

- The normalization parameters are initialized with their default values each time a speaker change is detected.
- The first utterance of the current speaker is decoded with some general adaptation parameters.
- The resulting hypothesis is used to compute a Viterbi alignment, which is then used to adapt the normalization parameters for VTLN and cMLLR as described above. The adaptation parameters for cMLLR have to be estimated after VTLN optimization. CMS and CVN are applied after VTLN.
- The updated normalization parameters are used for decoding the next utterance. The first utterance is also re-decoded.
- Depending on the situation, incremental adaptation either stops once a sufficient number of frames has been collected for robust estimation, or it continues indefinitely and a history weighting factor is applied instead. The second method is especially advantageous if the channel changes over the time and far-field microphones are used for recording.

As can be seen in Table 5.1 using either incremental cMLLR or VTLN leads to almost the same improvements in WER, except for *Lectures*. This can be explained by the fact that the speaker's VTLN warping factor is almost identical to the default one, i.e. no warping. As can also be seen, applying both VTLN and cMLLR at the same time gives some further improvements. Note that the results when applying both VTLN and cMLLR are identical to the "FC-MMIE" results from Table 4.7. To sum up, a relative gain of more than 12% can be achieved on *Lectures* by using online adaptation, and on all tasks the difference in WER for VTLN-AM and cMLLR-AM is insignificant.

Another insight into the behavior and improvements of online adaptation is shown in Figure 5.1. On the right axis, the average WER over time, using incremental adaptation during decoding is plotted. From this function, it can be seen that the WER decreases continuously over time, although there seems to be a slight increase in WER at the end. A possible explanation for this phenomenon could be the increasing fatigue of a speaker leading to more sloppy speech at the end of a lecture. On the left axis, the WER ratio between VTLN-AM and cMLLR-AM is plotted as a function of the amount of speech used so far for incremental adaptation. For both acoustic

	Dev		Eval	
	VTLN-AM	cMLLR-AM	VTLN-AM	cMLLR-AM
		RT-06S		
no adaptation	33.3%	33.0%	42.1%	42.6%
incr. VTLN	28.9%	29.0%		
incr. cMLLR	29.1%	28.7%		
both	27.9%	27.7%	35.6%	35.6%
		TC-STAR-07		
no adaptation	20.3%	21.1%	18.3%	18.7%
incr. VTLN	16.6%	16.9%		
incr. cMLLR	16.5%	16.7%		
both	15.8%	15.7%	14.3%	14.3%
		Lectures		
no adaptation	16.4%	16.1%	19.9%	19.8%
incr. VTLN	15.8%	15.6%		
incr. cMLLR	14.5%	14.5%		
both	14.3%	14.1%	15.7%	15.5%

Table 5.1: Comparison of the improvements due to incremental adaptation during decoding for systems using a VTLN or a cMLLR acoustic model.

models, VTLN and cMLLR parameters were updated incrementally on a per-utterance basis. For ratios smaller than one, VTLN-AM performs better than cMLLR-AM and for values grater than one vice versa. It should be noted that a difference of 0.01 in the WER ratio correspond to an absolute difference of about 0.15 in WER. From the ratio function over all lectures in *lectDev* it can be seen that a sufficient amount of speech for adaptation must be available before cMLLR-AM outperforms VTLN-AM. According to the figure, this happens at about 10-15 minutes of speech. By analyzing the results in more detail, it can be seen that this behavior is not true for all lectures. Mostly responsible for this behavior is *t012* for which cMLLR-AM is considerably better almost from the very beginning. For all other lectures, this is not the case. Furthermore, it can be observed, that there is a high fluctuation in the ratio when the amount of adaptation data is small.

One might assume that an optimal strategy would be to switch the acoustic models at a specific point of time, i.e. in this case after 10-15 minutes. But due to the inconsistent behavior of the lectures in *lectDev* and due to the insignificant difference of the final WERs for VTLN-AM and cMLLR-AM, no significant improvement could be achieved in this way.

Another strategy is to start each lecture with some pre-computed adaptation parameters. While this is relatively easy for VTLN, because of the smaller number of paramters to estimate, it is more complicated for cMLLR. In this case, a common technique is to cluster the speakers in the

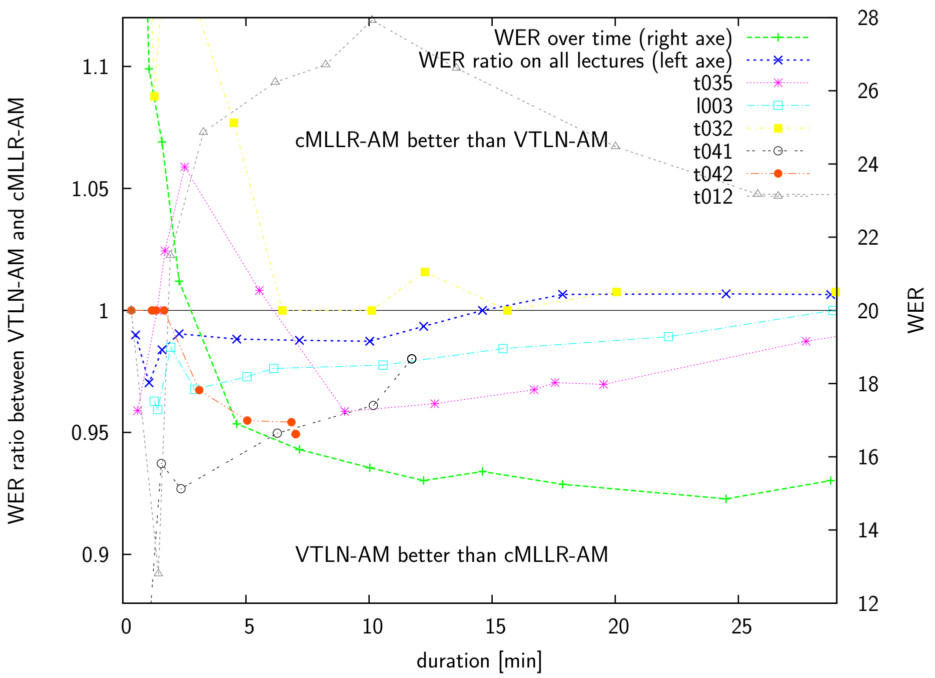


Figure 5.1: The right axis is for the average WER by using incremental adaptation plotted as a function of amount of data available for estimating the adaptation parameters. The left axis is for the evolution of WER ratios between VTLN-AM and cMLLR-AM as well as a function of the amount of adaptation data. All results are achieved on *lectDev*.

training data into several classes and to compute for each class adaptation parameters over all speakers in that class. After decoding the first utterance, the resulting hypothesis is used to select the best class. For each class the likelihoods are computed using a forced Viterbi alignment with the given hypothesis. After selection, the pre-computed parameters are mixed with the current estimate using a history weighting factor. For decoding the first utterance, some default parameters obtained either on the full training data or on a representative subset can be used. For clustering the speakers of the training data, two different techniques were explored (the number of classes where determined empirically beforehand to be 25):

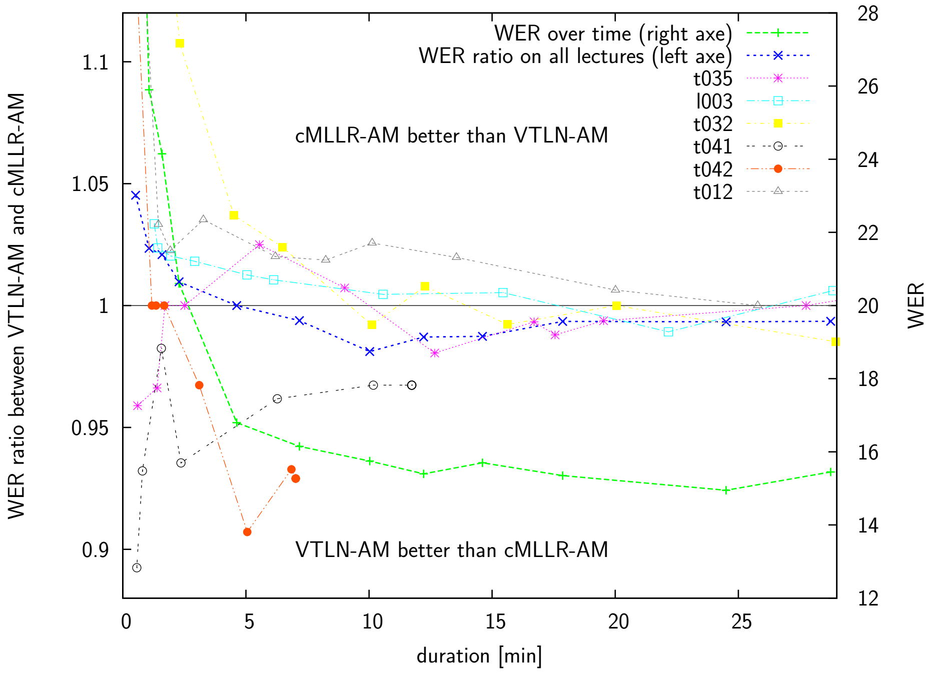
- The VTLN parameters were used to cluster all speakers in the training data into 25 classes. For decoding the first utterance, the parameters of the class of the default warping factor of 1.0 were used.

- The similarity between adaptation parameters (i.e. for cMLLR the rotation matrix and translation vector) are used to cluster all speakers in the training data into 25 classes. Similar to [MOS06], this was done by (1) vectorizing for each speaker the rotation matrix by appending all matrix columns to the translation vector, (2) normalizing each dimension over all speakers to have zero mean and unit variance, (3) using principal component analysis to reduce the dimensionality of the vectors, and (4) clustering the vectors of all speakers into 25 classes using k-means. For the third step, different dimensionality reduction methods were compared.

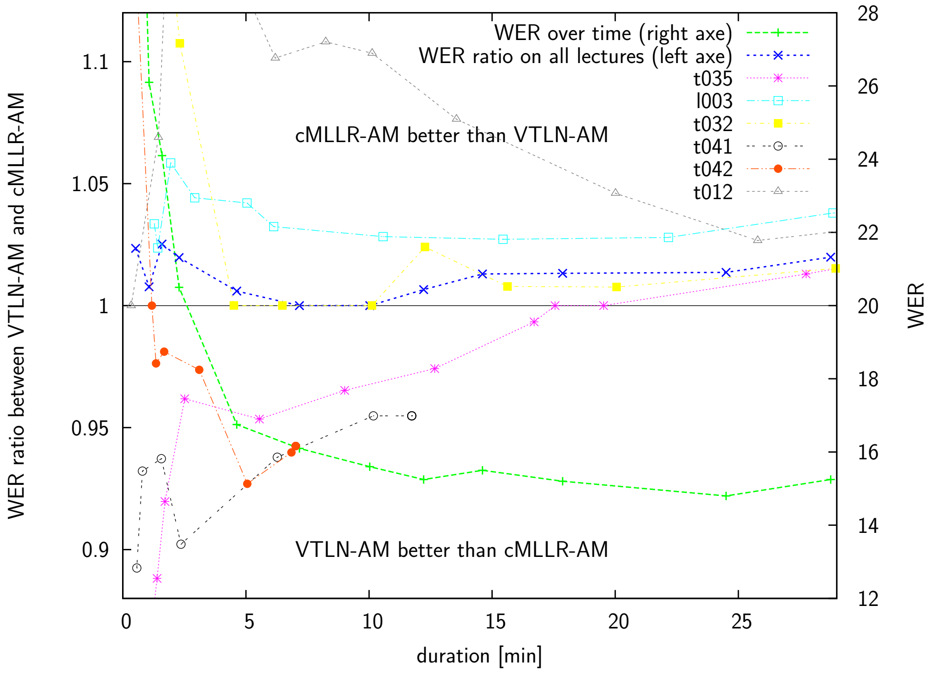
When comparing the WER ratio on all lectures between Figure 5.1, where no pre-computed adaptation parameters were used, and Figure 5.2, it can be seen that for both clustering techniques the performance on the first few utterances improved. This is recognizable in the vertical shift of some of the curves towards the top. However, this is not true for all lectures. But overall it can be observed that using the similarity of cMLLR parameters as clustering criterion performs better than using VTLN parameters, and compared to not using any pre-computed adaptation parameters at all, the overall WER could be reduced by 1.3% on *lectDev*.

5.4 Offline Adaptation

In contrast to online adaptation, offline speaker adaptation is performed only prior to decoding. Therefore, in addition to VTLN and cMLLR, also more computationally expensive techniques such as model space MLLR or MAP can be used. In the following experiments, the improvements obtained using supervised or unsupervised adaptation with different amounts of adaptation data of the same speaker will be analyzed. For this purpose, utterances from the *lectOther* (see Table 4.3) set were randomly selected across the different



(a) The adaptation parameters were obtained for each cluster after clustering the training speakers according to their VTLN parameters.



(b) The adaptation parameters were obtained for each cluster after clustering the training speakers according to their similarity in cMLLR parameters.

Figure 5.2: Evolution of WER ratios for VTLN-AM and cMLLR-AM on *lectDev* using pre-computed adaptation parameters.

talks until a specific amount of speech was reached. This was done to reduce the influence of a specific lecture and the order in which the lectures were used for adaptation. Since the random generator was always seeded with the same value, a larger set always contains the utterances of the smaller set as a prefix. Altogether, almost 240 minutes of speech were available. Similar studies have been published on other tasks e.g. in [LW95].

For unsupervised adaptation, confidence-annotated hypotheses produced with the VTLN system described above, using incremental adaptation, were used. From the result of experiments in the past, using only words receiving a confidence value higher than 0.5 for adaptation was the best. For supervised adaptation, the manual reference transcripts were used. In both cases, adaptation parameters were estimated for VTLN, cMLLR, and model-based MLLR for each speaker in the following way:

1. The VTLN warping factor is estimated by optimizing a maximum likelihood-based criterion along fixed state alignments using a Brent search over an interval of warping factors.

2. A global cMLLR matrix is estimated along Viterbi alignments in the vocal tract length normalized feature space.

3. In this normalized feature space, the statistics for model space MLLR are collected using Viterbi alignment and are applied to the acoustic model.

4. Steps 1 - 3 are repeated a second time in the already normalized feature space and with the MLLR-adapted acoustic models to improve the adaptation parameters and models.

As can be seen in Figure 5.3, for both supervised and unsupervised adaptation, the WERs decrease as the amount of adaptation data increases. The difference in WER between supervised and unsupervised adaptation grows as more adaptation data becomes available. Furthermore, the additional gain due to incremental online adaptation using the pre-computed offline adaptation parameters as a starting point can be seen as well. At a reasonable amount of adaptation data of 15 minutes — a common talking time of a single talk, a relative improvement in WER on *lectDev* of almost 5.7% after supervised adaptation and 2.1% after unsupervised adaptation could be achieved. When using the full set, the relative improvement increases to 12.8% after supervised adaptation and to 10% after unsupervised adaptation. In another experiment, it was analyzed whether the difference in WER between supervised and unsupervised adaptation can be preserved when the adaptation parameters and acoustic models are seeded with that of the supervised adapted system. Starting from a supervised adapted system using 15 minutes of speech data, unsupervised adaptation was performed. Unfortunately, the difference in WER could not be preserved, as can be seen

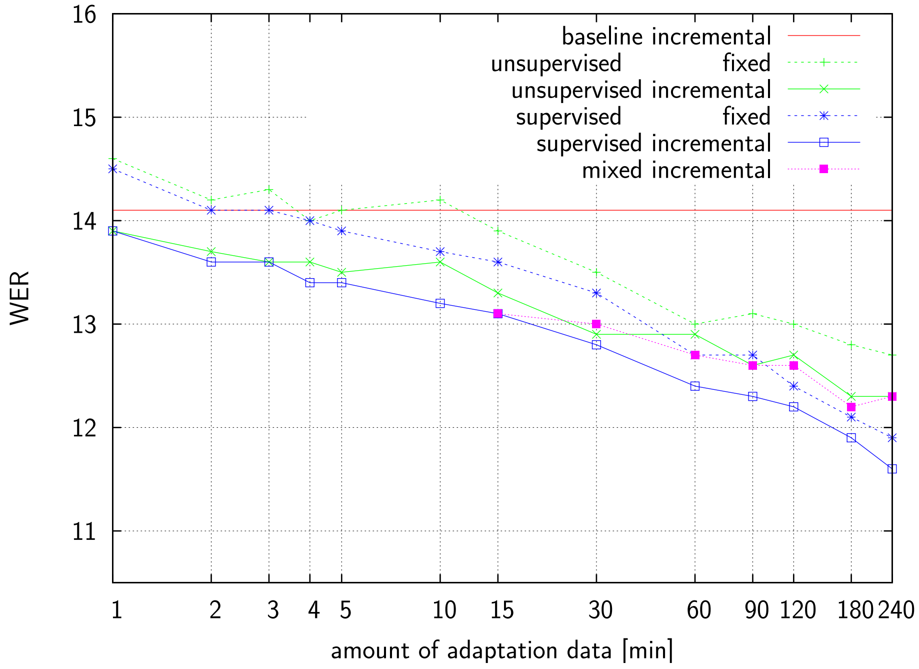


Figure 5.3: Comparison of *supervised* and *unsupervised* adaptation with respect to the amount of available data. For the curves marked with *incremental*, online adaptation was performed during decoding, while for the curves marked with *fixed* the pre-computed offline adaptation parameters were used instead and no online adaptation was performed. For the *mixed* curve, the unsupervised adaptation was seeded with a supervised adapted system.

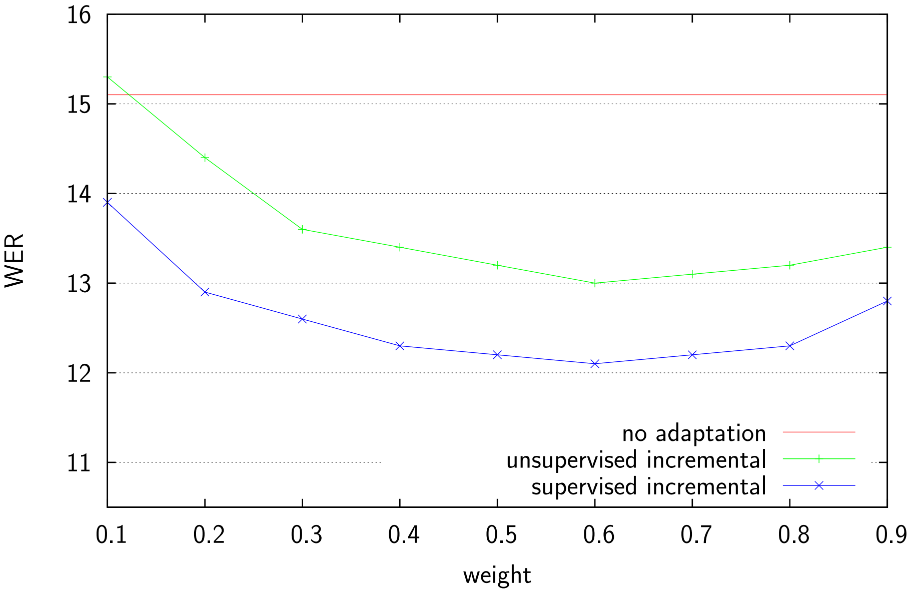


Figure 5.4: MAP with incremental adaptation during decoding using 240 minutes of adaptation data.

from the *mixed* curve in Figure 5.3. Another interesting observation from the experiments is that, when using no online adaptation during decoding, a system adapted with a small amount of adaptation data performs actually worse when compared to an unadapted system using online adaptation. With online adaptation during decoding, even a small amount of adaptation data results in an improvement in WER for the adapted system.

Due to the way the adaptation parameters are estimated and the Gaussian mixture models are updated, MAP is more sensitive to the amount of adaptation data than model-based MLLR. As can be seen in Figure 5.4, supervised adaptation is always about 1% absolute better than unsupervised adaptation and the optimum weight is around 0.6. But when comparing these results, which were achieved by using the whole 240 minutes of adaptation data, with those presented in Figure 5.3 it can be seen that MAP performs significantly worse than MLLR.

5.5 Conclusion

In this Chapter, different speaker adaptation techniques were compared with respect to their improvement in WER on lectures. For online adaptation using VTLN and cMLLR, a relative improvement of about 12% could be achieved compared to using no adaptation at all (Table 5.1). In another comparison, the improvement of online adaptation over time was analyzed between a system using an acoustic model trained with VTLN and another one trained with cMLLR in addition to VTLN. From Figures 5.1 and 5.2, it was seen how important pre-computed adaptation parameters are, especially for the first few utterances, if a cMLLR trained acoustic model is used for decoding. In advance to decoding, a set of adaptation parameters was obtained by clustering the training speakers, from which appropriate entries were selected during decoding. Two different speaker clustering techniques were developed, one using VTLN parameters and the other one using the similarity between cMLLR parameters. It was shown that significant improvements on the first few utterances could be achieved for both.

Furthermore, it was shown how available audio data of the same speaker can be used for supervised and unsupervised adaptation using model-based MLLR (Figure 5.3). In this context, it was shown that the results obtained with a supervised adapted acoustic model can also be improved using online adaptation. A relative improvement of about 3-5% could be obtained almost independently of the amount of data used for supervised adaptation. In addition, it was seen that the WER can be significantly improved even with a few minutes of data used for supervised or unsupervised adaptation. A relative improvement of about 5.7% and 7%, for unsupervised and supervised adaptation, respectively, compared to an unadapted system can be expected with about 10-15 minutes of audio data of the same speaker – a common speaking time for a talk. When using all available data (almost 4 hours), a relative improvement of 12.8% and 17.7%, for unsupervised and supervised adaptation, respectively, over the baseline of 14.1% was obtained, i.e. a final WER of 12.3%, respectively 11.6%.

Figure 5.5 compares the WER improvements due to the different adaptation techniques and the amount of data used to estimate the adaptation parameters on the evaluation data. Starting with a system using no adaptation at all, the WER could be reduced by 21% to 15.5% when using online adaptation (*incr-SA*). When 15 minutes of audio data of the same speaker becomes available, the WER could be further reduced by 4.5% for unsupervised (*unsup-incr-15*) and by 7.1% for supervised adaptation (*sup-incr-15*), in addition to online adaptation. Using all available data results in an improvement of about 11% for unsupervised and 16% for supervised adaptation, i.e. a final WER of 13.8%, respectively 13.0%.

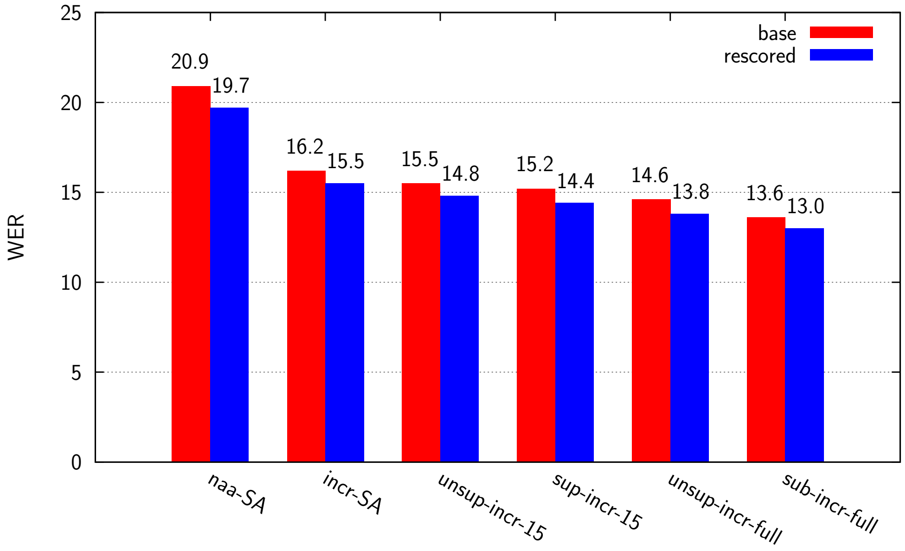


Figure 5.5: Comparison of the different speaker adaptation techniques on the evaluation data.

Chapter 6

Topic Adaptation

Depending on the amount of information available prior to simultaneous translation, different techniques can be applied to adapt the models of the system towards a specific speaker, topic, or even a special talk. Speaker adaptation was already discussed in the preliminary chapter. This chapter will concentrate on describing the topic adaptation framework. As in the previous chapters, the descriptions and experiments focus more on the adaptation of the speech recognition component of the system, i.e. the language model and vocabulary, rather than the translation component. Nevertheless, at the end of this chapter, it is shown that some adaptation techniques can easily be applied to the target language model in the translation component as well.

6.1 The Adaptation Framework

Figure 6.1 gives an overview of the adaptation framework developed in this thesis. This flowchart is processed by the lecture translation system prior to each lecture.

- If neither the lecturer nor the title or topic of a lecture are known in advance, the system loads general speaker-independent acoustic, language, and translation models.

- If only the name of the speaker is known, and the speaker has already given a lecture on which the system has adapted its models, the system is able to load speaker-adapted acoustic models. Since the topic is unknown, it has to load general adapted language and translation models. If there is no information about the speaker stored in the database, speaker independent models have to be loaded. In both cases, the information about the speaker can be used to query the Internet for previously given lectures or other publications by the same speaker to adapt the language and translation models.

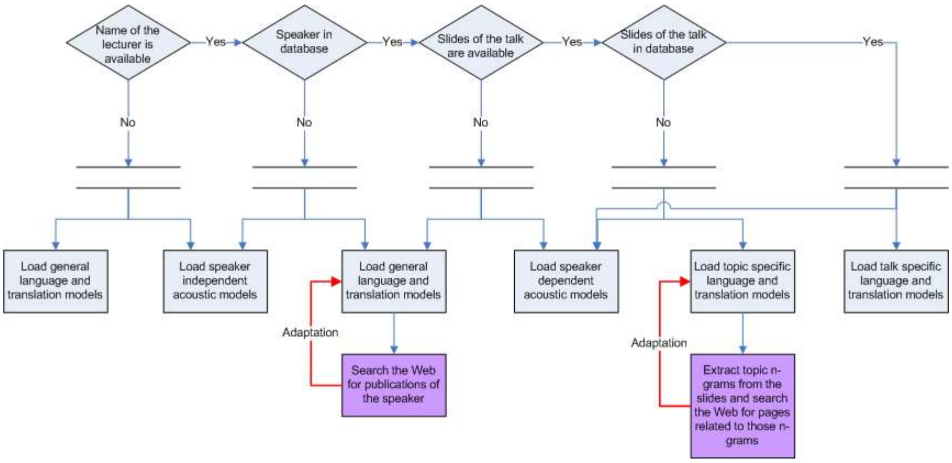

Name of the lecturer is available
Yes
Speaker in database
Yes
Slides of the talk are available
Yes
Slides of the talk in database
Yes
No
No
No
No
Load general language and translation models
Load speaker independent acoustic models
Load general language and translation models
Load speaker dependent acoustic models
Load topic specific language and translation models
Load talk specific language and translation models
Adaptation
Search the Web for publications of the speaker
Adaptation
Extract topic n-grams from the slides and search the Web for pages related to those n-grams

Figure 6.1: The adaptation framework. Flow-chart of the different adaptation granularities in a lecture translation system.

- If the title or the slides of the talk are additionally available in advance, this information can be used to search the Internet for even more specific material. Therefore, topic-related keywords or n-grams have to be extracted from the title or slides.

The more information about a speaker or topic is known in advance, the better the models can be adapted and the better the system might perform ultimately. The information can either be given manually or retrieved automatically from the Internet. For adaptation, the material collected must first be filtered and normalized in order to interpolate it with the other more general background models. Although not shown in the figure, it also makes sense to adapt the vocabulary of the speech recognizer by extending it with missing topic words or proper names. Section 6.4 deals with this issue separately. The next Section describes techniques for language model adaptation.

6.2 Language Model Adaptation

The main adaptation technique used is the so-called model interpolation, in which several language model components LM_i are linearly interpolated to form a mixture model. The probability for a word w following the word history h is then computed as follows:

$$P(w|h) = \sum_{LM_i} \lambda_i P_i(w|h), \quad \text{with} \sum_i \lambda_i = 1 \tag{6.1}$$

The mixture coefficients or interpolation weights λ_i may be a function of the word history h and are typically estimated with the help of the EM algorithm so that the perplexity of the mixture model on some held-out data is optimized. Language model adaptation is performed by updating the interpolation weights with the help of additional held-out data and/ or by exchanging or adding some language model components which better cover the desired topics. Adding new components can be done either by interpolating them with the already existing language model, or by re-estimating all mixture coefficients. While in the first case the existing (background) language model is kept unchanged, in the second case the interpolated language model has to be computed from scratch. The first method is preferred because of its smaller computationally effort, and used for most of the experiments in this thesis, but as will be seen later, better results can be obtained by using the second approach.

Other techniques used for language model adaptation include dynamic cache language models [KdM90], minimum discriminant information (MDI) [Fed99], or latent semantic analysis (LSA) [Bel00], but are not further investigated in this thesis. A more detailed overview of statistical language model adaptation techniques can be found in the journal article by Bellegarda [Bel04].

6.2.1 Useful Data for Adaptation

But what kind of data is useful for adaptation and where can that data be obtained? This section deals with these two questions and describes the problems and advantages of different types of data.

Manual or Automatic Transcripts

The best data for language modeling for speech recognition are transcripts of speech data. This data is preferred over text data because spoken language differs in style from text, due to spontaneous effects like word reorderings, word repetitions, breaks, hesitations and other disfluencies. Since speech recognizers have to recognize word-by-word, well-estimated transitions covering these kinds of disfluencies are essential.

Transcribing speech data can be done manually, or automatically if a good enough speech recognition system is available. Initially during system development, this is not the case, and, due to recognition errors manual transcripts are usually preferred over automatic ones. On the other hand, manually transcribing data is very costly and many occurrences of disfluencies further complicate this process. Therefore, different approaches to spontaneous language modeling have already been investigated in the literature. A good introduction to the problem can be found in the article by Stouten [SDMW06].

Often, the estimation of the probabilities for inserting disfluencies is treated separately from statistical n-gram language modeling. For example, in [SS96], a so-called hidden event language model was used. In [HWM03], a weighted finite state transducer for speaking style translation was proposed instead. However, the success of such approaches, in comparison to modeling them directly in the n-gram language model, depends on the transcribed spontaneous speech already available.

The recognition system used in this thesis models some disfluencies, such as word repetitions or breaks, directly, with the help of the language model. In contrast, so-called filler words which are allowed to occur between any other two regular words are handled differently. Their language model probabilities are not computed by the language model. Instead, a fixed filler penalty optimized on some held-out data is used. The category of filler words covers hesitations as well as human and non-human noises.

Topic-Related Text Data

For topic adaptation, data related to a particular topic is necessary. Again, transcripts are preferred over text data, but are more difficult to find than spontaneous speech transcripts. In the literature, one can distinguish between approaches for selecting data related to a particular topic out of a large corpus [MBH99], approaches for further collecting related text data by querying a search engine and retrieving resulting web pages from the World Wide Web [BOS03, WH06, SGN05], or a combination of both. Using the huge amount of data available in the Internet has increased in popularity over the last years because of the higher quality of search engines. Written data which is similar to spontaneous speech can be found in many chat rooms [BOS03].

As already introduced during the explanation of the adaptation framework, the approach used in this thesis is to query the Internet. Therefore, queries have to be generated for retrieving relevant documents by using web search engines.

6.2.2 Query Generation using TF-IDF

The procedure for query generation presented here was developed for the RT-06S evaluation. For web text collections used for training the *LM6* language model, two different web query strategies were employed. For the *UKA-L* collection, the same web text collection framework as proposed in [BOS03] was followed, where frequently spoken 3-grams and 4-grams from the target task training data are combined to form queries. For the other collections, a more sophisticated approach was used. Frequent n-grams from different lecture or conference meeting transcripts were combined with topic bigrams from the conference proceedings *PROC* to form queries. The goal was to obtain text reflecting a broad variety of topics, some of which are not represented in the training set, as well as data covering the desired speaking style. Note that the n-grams were put in quotes before being transmitted to the search engine (Google). All in-house collected web data was perplexity filtered to roughly match the size of the *UW-M* web collection.

For computing topic phrases, a technique known from the field of Information Retrieval and similar to [RS02] was used. With *term frequency - inverse document frequency* (*tf-idf*), a weight is defined which is a statistical measure of how important a word is to a document in a collection of documents. The weight w_k for a word or term k is defined as

$$w_k = \frac{g_k}{\sqrt{\sum_k (g_k)^2}}, \tag{6.2}$$

$$g_k = tf_k \log\left(\frac{N}{f_k}\right) = tf_k \cdot idf_k, \tag{6.3}$$

where tf_k is the term frequency, i.e. the number of occurrences of the term k in the current document, N is the number of documents in the collection, and f_k is the number of documents which contain the term k. The term weight g_k increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the collection. To obtain w_k, the term weight g_k is normalized to the length of the document.

In contrast to the approach used in [RS02], which computes *tf-idf* weights for words only, bigrams are used as single terms. Since not all bigrams are relevant, a selection algorithm was developed.

1. For each document in the proceedings data *PROC*, the *tf-idf* weight is computed for all bigrams.

2. All weights but the top 10% are zeroed.

3. The obtained weight vectors are averaged over the collection, and the top 1400 bigrams excluding any with stop-words or numbers are selected.

4. The topic bigrams are mixed randomly with general phrases until the desired number of queries, i.e. in this case 14k are generated.

The second step reduces the list to the most relevant bigrams. In the third step, all non-topic bigrams are removed with the help of a stop word list containing 1500 high frequency function words. The last mixing step ensures that the data collected from the Internet also matches the speaking style. The top ranked topic bigrams are, e.g.

> language model, speech recognition, time frequency, speaker recognition, vocal tract, clean speech, channel estimation, audio visual, space time, acoustic models

A frequency-based approach would select the following top ranked bigrams instead:

> language model, speech recognition, error rate, acoustic models, natural language, proposed method, training data, clean speech, acoustic model, broadcast news

While some of the bigrams between both approaches are identical, it can be seen that by using *tf-idf* more specialized bigrams are preferred over the more general ones.

For the adaptation framework, the topic information has to be extracted for a single talk only, instead of a collection of documents. However, *tf-idf* weights can only be computed with respect to a collection of documents. Therefore, the data from which the components for the baseline (background) language model were built is used as a background corpus.

The above-presented algorithm has the problem that for some n-grams, *tf-idf* weights could not be computed because the corresponding n-gram in the background corpus was missing. Furthermore, the hard thresholds defined in the selection algorithm makes it unsuitable for changing data sizes. Therefore, the selection algorithm has to be further refined, and, instead of bigrams only, also trigrams were considered. Given some data related to the current talk, like presentation slides or publications the topic n-grams are selected according to the following heuristic:

1. Compute bigram and trigram *tf-idf* weights on the relevant data with respect to the background corpus.

2. All n-grams with a weight higher than 10% of the highest weight are selected.

3. In addition, all n-grams with no *tf-idf* weight and a higher occurrence count than the square of the average occurrence n-gram count are selected.

The count and *tf-idf* thresholds are defined for each n-gram category separately.

6.3 Language Model Adaptation Experiments

In the following experiments, the different adaptation levels of the proposed adaptation framework are evaluated with respect to improvements in speech recognition.

In Section 6.3.1, a detailed description of differently adapted language models used as baseline language models for the following experiments are given. Section 6.3.2 analyzes to which extent the language model can be adapted, if the speaker's identity is known. Therefore, the homepage of the speaker as well as the speaker's publications are retrieved from the web. In Section 6.3.3, this data was used to retrieve other related data, similar in topic, and the relevance of this data is analyzed in more detail.

So far, only data related to the interests of the speaker was collected. In the next step, it was analyzed if the adaptation could be improved by data related to the talk or lecture the speaker is given. Therefore, in Section 6.3.4, information on the presentation slides was used to retrieve relevant data from the web, and the results when using this data for language model adaptation prior to decoding are presented.

Sections 6.3.5 and 6.3.6 analyzes the improvements possible if more data from the speaker becomes available such as recorded lectures. Depending on the type of the transcripts available, manually transcribed or automatic, the improvements in WER by using this data for LM adaptation is studied.

6.3.1 Baseline Language Models

We analyze in more detail the language model used for the baseline ASR system. As described in Section 4.3.6, this language model (*LM6*) is a mixture language model consisting of nine different language model components; it evolved over several evaluations [FKPW06, FIK+06]. The main improvements were made prior to the RT-06S evaluation when a huge amount of additional data was collected from the Internet with the technique described in Section 6.2.2.

For the experiments within this chapter, *LM6* was modified to better handle disfluencies. As described above, instead of estimating their transition probabilities by the language model, a fixed penalty was used instead. This was done for only those disfluencies which seem to occur independently from their surrounding word context. As a result, the perplexities as well as the WERs given in this chapter differ from those presented in Section 4.3. Due to this change, the PPL improves by 17% on *RT-06Sdev* and 8% on *lectDev*, and the WER improved by 3.2% and 2.8%, respectively. The improvement is not only because of the changed modeling; the penalty applied is also much smaller than compared to the average language model probability estimate of a disfluency transition in *LM6*, making it more likely for disfluencies to be recognized. The same normalization was applied to all other LMs in this chapter. It should be noted that, throughout all language model adaptation experiments, the language model vocabulary was kept fixed to the one described in Section 4.3.5.

Figure 6.2 shows the mixture coefficients for *LM6*, and Table 6.1 gives the corresponding perplexities and word error rates on different development and evaluation sets. For *LM6*, the mixture coefficients were tuned on a held-out set related to the RT-06S evaluation and different from the development and evaluation sets. As can be seen, the most relevant data are the transcripts of recorded seminars *SMNR*, followed by the proceedings data. The fact that the proceedings data are only second in importance shows the technical relatedness of the held-out data.

As mentioned above, due to the use of language model components built

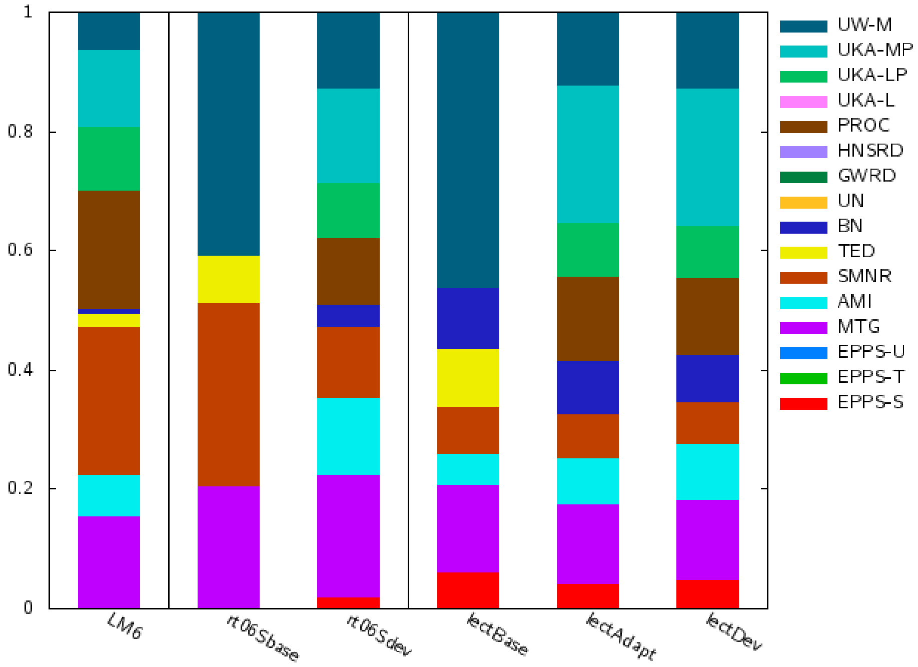


Figure 6.2: Mixture coefficients of the training corpora tuned on different development sets. Components for which the coefficients are smaller than 0.2 are not shown in the figure and also not used for the estimation of the interpolated LM.

	RT-06Sdev		lectDev	
	PPL	WER	PPL	WER
LM6	127	27.0%	150	13.9%
rt06Sbase	136	28.2%	162	14.5%
rt06Sdev	124	26.9%	145	
lectDev	127		144	13.5%

Table 6.1: In- and across-domain perplexity (PPL) and word error rate (WER) comparison of mixture LMs. *rt06Sbase* was tuned on RT-06S held-out data, *lectBase* on the held-out set *lectOther*, and *rt06Sdev* and *lectDev* on the corresponding development sets instead.

from proceedings data, as well as from large amounts of web data related to technical presentations, *LM6* is already highly adapted towards the lectures which are considered in this thesis. As a result of this, the potential of the developed adaptation framework is hard to assess because obtained improvements on top of the results achieved with *LM6* – if any – are rather small. To overcome this problem, a new unadapted baseline language model *rt06Sbase* was created, for which the allowed corpora were restricted to all audio transcripts, as well as the text data not collected in the context of *RT-06S*, i.e. *UW-M*, *HNSRD*, *GWRD*, *UN*, and *BN*. As a held-out set for computing the mixture coefficients, the RT-06S held-out data was used. This corresponds to the realistic situation of applying the lecture translation system, for the first time for a specific speaker and a particular topic, given that the system had been already used in a similar situation for another speaker. Besides some related transcripts and additional held-out data, no other corresponding data is available.

As can be seen in Figure 6.2, the most relevant corpora for *rt06Sbase* are *UW-M*, *SMNR*, *MTG*, and *TED*. And Table 6.1 shows that on *RT-06Sdev*, a perplexity of 136, and 162 on *lectDev* can be achieved. The WERs obtained on *RT-06Sdev* and *lectDev* are 28.2%% and 14.5%, respectively.

For comparison reasons two other language models were created. The *rt06Sdev* language model was tuned with respect to the *RT-06Sdev* data, and the *lectDev* language model with respect to the *lectDev* data, rather than with respect to the corresponding held-out data sets. Compared to *LM6*, some differences can be observed. The proportions between different components are modified, the *TED* language model is not important anymore, and the *EPPS-S* language model is added with a small weight. While for *rt06Sdev* the meeting transcripts are the most important, for *lectDev* the web data based on meeting transcripts and proceedings is the most relevant. Due to this difference, the perplexities on the corresponding data set are also reduced by 3 and 6 for *RT-06Sdev* and *lectDev* respectively, relative to *LM6*. Only a slight WER reduction for *rt06Sdev* of 0.4% to a WER of 13.5% was observed.

The language models *lectBase* and *lectAdapt* will be described later in Section 6.3.5. To reduce the size in memory, all language models were pruned with an entropy-based criterion using a threshold of 10^{-9} [Sto98].

6.3.2 Using the Speaker's Identity

With the help of the speaker's identity, the Internet can be queried for additional information about the speaker. Ideally, the homepage, along with the speaker's publications can be found. Although querying a search engine with the speaker's name is possible, a better method is to search manually for the necessary information, since the name might not be unique. If the homepage is found, a web crawler can be used to retrieve the necessary pages recursively, i.e. together with linked pages up to a specified linking depth. All the found pages as well as the publications can now be filtered, normalized and merged. Note that for this adaptation level, no topic n-grams have to be extracted.

Filtering removes unnecessary information such as HTML code, tables, formulas, special characters, etc. After this, sentence boundaries or paragraphs are detected. Additional filters are applied to remove sentences, paragraphs or documents if they have a worse ratio of real words to out-of-vocabulary words, of numbers to words, or of number of characters to words. Out-of-vocabulary words are identified with the help of a large background vocabulary with about 500k entries. The normalization step writes out abbreviations and numbers, tags acronyms, and removes punctuation. The normalized pages and documents can now be merged into a single corpus.

Experiments

The following experiments were conducted on the *lecture* task as well as on the *RT-06S* data. Due to the lack of held-out and evaluation data with additional speaker information, it was necessary to partition the *RT-06Sdev* set randomly into a development (*RT-06Sdev-dev*) and evaluation (*RT-06Sdev-eval*) set. All word error rates are given after language model rescoring, with the language model parameters optimized on the development data.

Data: Table 6.2 gives an overview of the amount of data collected per speaker as well as the adaptation results. As mentioned above, *rt06Sbase* was used as a baseline, and as the background language model for interpolation with the foreground language models built from the data collected. Since some of the speakers are students, no additional information or publications were found about them on the Internet, and the data found about their supervisor was used instead. As a result, for two speakers with the IDs *evsmbd* and *robdkz*, as well as for the two speakers *pgaftk* and *qgzhyn*, the same data is used. As can be seen in the second column of Table 6.2, the amount of data found per speaker varies largely from about 3k to 555k.

The relevance of the collected data: For measuring the relevance of the collected data with respect to different speakers, a language model build on a speaker's data only. The perplexities were computed on all speakers' reference transcripts of the development set and compared. Table 6.3 shows the results. As can be seen, the perplexities for the matched condition are usually amongst the top two lowest for a single speaker. This is not the case for only those speakers for which the data used is identical to that of another speaker, and for *evsmbd*. An analysis of the transcripts for *evsmbd* shows that this speaker employs more general language than others.

speaker ID	# words	PPL-b	WER-b	weight	PPL-i	WER-i
RT-06Sdev development set						
bpdnpq	17,409	165	28.6%	0.27	118	25.1%
evsmbd	155,378	101	14.9%	0.14	97	13.4%
ksb	425,853	154	24.7%	0.33	124	23.8%
mlspsf	3,280	103	21.0%	0.05	101	20.4%
ouarkw	234,014	119	28.2%	0.17	110	26.8%
pgaftk	5,100	132	25.6%	0.02	132	25.9%
qgzhyn	5,100	157	23.3%	0.09	145	24.1%
robdkz	155,378	117	30.3%	0.10	104	29.5%
spipaa	201,074	182	22.4%	0.37	141	20.0%
Overall	1,202,586	135	25.1%	0.17	122	24.2%
+ threshold				0.23		24.0%
RT-06Sdev evaluation set						
jjabpr	155,378	166	41.3%	0.17		40.5%
kesysg	188,524	143	29.0%	0.17		27.7%
kuvvqe	336,840	137	44.7%	0.17		42.8%
ojtlvn	2,925	157	25.6%	0.17		26.5%
owm	555,209	130	23.6%	0.17		22.5%
pgv	103,269	124	23.4%	0.17		22.3%
ristxa	74,375	123	26.5%	0.17		27.2%
tbvstt	5,344	126	32.2%	0.17		32.1%
xboxkz	20,468	129	29.8%	0.17		29.6%
Overall	1,442,332		30.3%	0.17		30.0%
+ threshold				0.23		29.9%

Table 6.2: Amount of data collected per each speaker on the *RT-06Sdev* set together with the mixture weights and resulting perplexities (PPL) and word error rates (WER). PPL-b and WER-b are the results computed with the baseline language model *rt06Sbase*, and PPL-i and WER-i are the results after language model adaptation using the collected data. WERs are given after language model rescoring.

data\ref	bpdnpq	evsmbd	ksb	mlspsf	ouarkw	pgaftk	qgzhyn	robdkz	spipaa
bpdnpq	**440**	**584**	1794	1486	1562	2242	1892	1342	2068
evsmbd	**406**	**320**	1030	904	987	1480	1209	1132	1284
ksb	629	**184**	**234**	368	572	694	528	769	649
mlspsf	2096	**1166**	2264	**1029**	2995	3001	2837	3205	3461
ouarkw	322	**179**	596	539	**450**	1023	712	772	782
pgaftk	2902	**1399**	2436	1752	3482	2569	**1480**	4142	3495
qgzhyn	2902	**1399**	2436	1752	3482	2569	**1480**	4142	3495
robdkz	**406**	**320**	1030	904	987	1480	1209	1132	1284
spipaa	654	**242**	459	335	467	604	545	572	**258**

Table 6.3: Perplexities of different speaker's data on different speaker's reference transcripts restricted to *RT-06S-dev-dev*. In bold, the two lowest perplexities per row are marked.

It could be shown, that the data retrieved for a specific speaker is correlated in terms of perplexity with the presentation of the speaker. In the next two experiments it is analyzed, if this relationship carries over to improvements in WER. In the first experiment (A), *lecture meetings* are explored, and in the second experiment (B), *lectures*.

(A) Lecture meetings: In the third and fourth column of Table 6.2, the perplexities (*PPL-b*) and word error rates (*WER-b*) achieved with the baseline language model are given. Overall, a perplexity of 135 together with a WER of 25.1% is obtained on the development set. For each speaker, the language model component built on the speaker's collected data is interpolated with the background language model. The optimized mixture weights obtained on the speaker's reference transcripts are given in the fifth column (*weight*), and the resulting perplexity on the same references in the sixth column (*PPL-i*). Due to the fact that speaker-dependent mixture weights cannot be estimated on the evaluation set, a single average mixture weight was computed. For the decoding experiments, this weight was used for the interpolation of the language model components for all speakers. This means that the word error rates given in the last column are obtained with the speaker adapted mixture models using a weight of 0.17 for the foreground language model component. As can be seen, the overall baseline WER can be improved by 3.5% to 24.2% on the development set. Although the improvement is not as large as on the development set, an improvement from 30.3% down to 30.0% could also be observed on the evaluation data. Another observation is that the adaptation typically works better for speakers for whom more data is available. Therefore, the result can be further improved slightly when the speaker-adapted language model is used only if the collected data amounts to more than e.g. 10,000 words. The WER could be improved to 24.0% on the development set and to 29.9% on the evaluation set.

Unfortunately, the random partition of the *RT-06Sdev* speakers into the two sets was not ideal. The development set is much easier to recognize than the evaluation set. Moreover, the mixture weights obtained on the development set are overestimated compared to those which would be achieved on the evaluation set. The average optimal weight on the evaluation set is 0.14 – much lower than the computed 0.23.

(B) Lectures: The above-described adaptation strategy was also applied to the *lecture* task. Since all data was recorded from a single speaker a single corpus with an amount of 607,097 words was collected. The language model component built on this data was then interpolated with a weight of 0.15 with the background language model *rt06Sbase*, with the mixture coefficient optimized on *RT-06Sdev-dev*. As can be seen in Table 6.4, the

ID	PPL-b	WER-b	PPL-i	WER-i	WER-b'	WER-i'
t035	164	14.7%	152	14.5%	14.7%	14.3%
l003	154	14.6%	150	14.3%	14.5%	14.3%
t012	168	17.4%	156	17.1%	17.3%	17.1%
t032	148	12.6%	142	12.4%	12.5%	12.3%
t041	196	14.3%	194	14.6%	14.2%	14.3%
t042	192	12.1%	175	12.5%	12.1%	12.1%
Overall	162	14.5%	156	14.3%		

Table 6.4: Word error rates and perplexities on *lectDev* before and after language model adaptation. While *WER-b* and *WER-i* are those obtained with the global best language model parameters, *WER-b'* and *WER-i'* shows the word error rates with the local best language model parameters. The mixture coefficient was optimized on the above introduced *RT-06Sdev* development set to 0.15.

overall perplexity was reduced from 162 (*PPL-b*) to 156 (*PPL-i*), and the word error rate from 14.5% (*WER-b*) to 14.3% (*WER-i*). However, this reduction is not obtainable for all lectures. Although the perplexity was reduced for all lectures, the word error rate for *t041* and *t042* increased. The reason for this behavior is due to the way in which the results after language model rescoring are obtained. The WERs presented in the *WER-i* column are those achieved with the language model parameters achieving the overall best WER. However, these language model parameters might not be optimal for all lectures. Therefore, in the *WER-b'* and *WER-i'* columns, the WERs achieved with the best language model parameters per lecture are given. As can be seen, these WERs correlate better with the presented perplexities. However, unless otherwise specified, the tables which follow give only *WER-b* and *WER-i*.

Conclusion

In this Section, it was shown that data retrieved from the homepage of a speaker together with the speaker's publications contains useful information relevant to a speaker's presentation. Using this data for language model adaptation by linear interpolation with a background language model improves perplexity as well as word error rate. On lecture meetings, the WER could be improved by 3.5% and 1% relative on the development and evaluation data, respectively. On lectures, the WER could be improved from 14.5% to 14.3%. Although consistent, the improvements are relatively small. Therefore, the next section investigates, if the results can be improved, if this data is used as a seed corpus for collecting additional data.

		Stopic		Sseed+Stopic	
speaker ID	# words	PPL-i	WER-i	PPL-i	WER-i
RT-06Sdev development set					
Overall		118	23.7	115	23.4
RT-06Sdev evaluation set					
jjabpr	9,343,834	147	39.5	148	38.6
kesysg	16,804,860	122	28.7	119	27.5
kuvvqe	26,717,728	135	43.2	134	42.4
ojtlvn	5,875,099	161	26.3	164	26.1
owm	26,518,836	136	23.1	137	24.1
pgv	10,196,691	113	22.1	107	21.4
ristxa	17,432,369	126	26.7	123	27.3
tbvstt	7,398,801	123	30.3	127	31.2
xboxkz	1,755,515	125	30.5	117	29.7
Overall			29.8		29.8

Table 6.5: Results for language model adaptation by interpolation of the background LM with the *Stopic* LM component only and with the *Sseed* LM component in addition. The results are given in perplexity and word error rate.

6.3.3 Speaker Related Web Data

For the experiments in this section, the data collected in the section above is used as a seed corpus for extracting topic n-grams, using the proposed *tf-idf* based query generation technique. The seed corpus or language model built using this corpus is referred to as *Sseed* and the corpus or language model built using the corpus collected with the help of *Sseed* is referred to as *Stopic* (with *S* for speaker). As above, this adaptation strategy is applied to both lecture meetings and lectures. The WERs achieved with two-fold and three-fold interpolation will be compared. in case of the two-fold interpolation, the *Stopic* LM is interpolated with the background LM alone, and, in case of the three-fold interpolation in conjunction with the *Sseed* LM.

Lecture meetings: For lecture meetings, the mixture coefficients for the two- and three-fold interpolation were optimized on the speaker's corresponding reference transcripts in *RT-06Sdev-dev* and averaged. The average mixture coefficient was used to compute the speaker-dependent language models for the development and evaluation sets. In the first case, a mixture coefficient of 0.35 for the *Stopic* LM was computed. In the second case, 0.26 and 0.11 were computed for *Stopic* and *Sseed*, respectively.

In Table 6.5, the amount of data automatically collected per speaker is given. It can be observed, that the number of words correlates with the number of words in the *Sseed* corpus, because the number of queries generated automatically depends on the size of the *Sseed* corpus. Furthermore, compared to Table 6.2, the perplexity and word error rate can be further reduced on the development set. On the evaluation set, the WER decreases only slightly for the two-fold interpolation with the *Stopic* LM to 29.8%, but no further when the *Sseed* LM is added. A correlation of the numbers of words collected and the WER achieved as compared to Table 6.2 was not observed.

Lectures: When using the same technique for the lectures an amount of 34,116,715 words was collected for the single speaker. Bigrams and trigrams were extracted from the *Sseed* corpus as search queries for the *Stopic* corpus. Altogether, 1242 bigrams and 1922 trigrams were extracted. The top ranked bigrams are:

> user registration, n-best lists, n-gram language, hand color, speech-to-speech translation, Toolkit JRTk, CallHome Spanish, human-robot interaction, speaker initiative, human-human communication, close-talking microphone, cross-language transfer

while the top-ranked trigrams are:

> Recognition Toolkit JRTk, GE EN JA, EN JA SP, CH GE EN, n-gram language models, discriminant analysis LDA, multimodal user registration, n-gram language model, vectorized nonterminal symbols, machine translation SMT, hand color model

As can be seen, several specialized n-grams are mixed with general ones. The n-grams similar to "GE EN JA" are common abbreviations for language pairs in machine translation. But also the lowest ranked n-grams seem to be relevant, as can be seen in the following trigrams:

> speech recognition error, extract signs robustly, recognition error rates, Ibis single pass, distant microphone conditions, bilingual Basic Travel, individual person activity, global STC transforms, Language Adaptive Acoustic, Sixth Framework Programme

The mixture coefficients were again optimized on the *RT-06Sdev-dev* development corpus. For the two-fold interpolation of the background LM with the *Stopic* LM, a coefficient of 0.31 for the *Stopic* was obtained, and, after additionally applying the *Sseed* LM, 0.08 and 0.24 for the *Sseed* and *Stopic* LM, respectively.

Conclusion

In this section, the *Sseed* corpus was used as a seed corpus for extracting topic n-grams, using the proposed *tf-idf* based query generation technique.

	Stopic		Sseed+Stopic	
ID	PPL-i	WER-i	PPL-i	WER-i
t035	147	14.5%	144	14.6%
l003	147	14.3%	145	14.4%
t012	156	17.1%	152	16.6%
t032	137	12.4%	136	12.1%
t041	181	14.6%	181	14.1%
t042	163	12.5%	160	11.9%
Overall	154	14.2%	151	14.1%

Table 6.6: Word error rates and perplexities on *lectDev* obtained with language models adapted by interpolation with the *Stopic* LM and with the *Sseed* LM in addition.

The resulting queries were used for retrieving additional data forming the *Stopic* corpus. Compared to the results of Section 6.3.2 perplexities as well as word error rates improved further when both corpora, *Sseed* and *Stopic* were used for a three-fold interpolation with the background language model *lectBase*. Compared to results obtained with the background language model, the WER was improved by 1.7% and 2.8% on lecture meetings and lectures, respectively.

So far, the experiments focused on general data related to the speaker and was not specific to a presentation. In the following section, it will be investigated, if data specific to a presentation can be retrieved from the web by using information extracted from the presentation slides.

6.3.4 Using Presentation Slides

In this section, additional information about the presentation is used for collecting data from the Internet. Examples of suitable information include presentation slides or manuscripts. Unfortunately, additional information in the form of presentation slides was not available for the *RT-06S* data; the following experiments were therefore carried out on the *lecture* data only.

As in Section 6.3.2, after describing the collected data per lecture in more detail, its relevance for the corresponding lecture is shown. The experiments are following afterwards. It will be investigated if (A) relevant data can be collected with the proposed *tf-idf* based query generation and web data collection strategy, (B) perplexities as well as word error rates improve when this data is used in conjunction with *Stopic*, and (C) by merging this topic-specific data into a single corpus to increase its relevance, further improvements can be achieved.

ID	# bigrams	# trigrams	# words
t035	221	88	5,957,056
l003	1150	388	24,266,155
t012	237	72	6,055,540
t032	317	433	9,943,847
t041	166	145	5,387,185
t042	324	174	10,438,130

Table 6.7: Number of queries extracted automatically from the presentation slides together with the number words collected per lecture on *lectDev*.

	Ltopic		Stopic+Ltopic	
ID	PPL-i	WER-i	PPL-i	WER-i
t035	154	14.6%	146	14.7%
l003	144	14.2%	144	14.4%
t012	164	17.3%	156	16.7%
t032	140	12.3%	137	12.2%
t041	178	13.8%	179	13.7%
t042	174	12.0%	163	11.9%
Overall		14.2%		14.1%

Table 6.8: Results achieved with language model adaptation using web data collected from presentation slides on *lectDev*. The mixture coefficients were optimized on the *RT-06Sdev* development set.

Data: Table 6.7 shows the number of queries extracted from the presentation slides and the amount of data collected from these queries per lecture. Again, the amount of data collected depends on the number of queries extracted. The language model or corpus extracted in this way is referred to as *Ltopic* (with *L* for lecture dependent).

(A) The relevance of the collected data: To verify the relevance of the collected data with respect to the different lectures, Table 6.9 compares the perplexities of different topic data collected for a specific lecture computed on different reference transcripts. Except for *t012* and *t041*, the corresponding reference transcripts can be found under the two lowest perplexities. In comparison with the *Stopic* data, it can be seen that the perplexity is not always lower.

In summary, the *tf-idf*-based query generation and web data collection strategy collects relevant data, with the relevance increasing as the corpus grows in size. The data retrieved for a specific lecture is correlated in terms of perplexity with the lectures reference transcripts. However, a large speaker-dependent corpus, general in topic (*Stopic*) may be often more suitable

data\ref	t035	l003	t012	t032	t041	t042
t035	**203**	296	258	**210**	243	240
l003	201	**199**	229	**182**	260	221
t012	**230**	265	262	**214**	271	260
t032	**195**	259	234	**186**	235	196
t041	**210**	296	266	**218**	222	240
t042	**219**	356	281	220	255	**203**
Stopic	**188**	202	**212**	**176**	238	**197**

Table 6.9: Perplexities of different *Ltopic* data collected for a specific lecture on different reference transcripts. In addition, the perplexity of the *Ltopic* data is compared with the *Stopic* data. In bold, the two lowest perplexities are marked per lecture or talk (row). For *Stopic*, the perplexity is marked bold if it is lower than the corresponding *Ltopic* perplexity.

especially if the topic dependent corpus (*Ltopic*) is relatively small.

In the next two experiments it is analyzed, if this relationship carries over to improvements in WER. In the first experiment (A) presentation specific data will be used, in the second experiment (B) the presentation specific data will be merged to a single corpus.

(B) Presentation specific data: For each *Ltopic* component, the mixture coefficients were optimized on the *RT-06Sdev-dev* set, separately. The mixture coefficient for *Ltopic* in the case of two-fold interpolation ranges from 0.06 to 0.12, and if the *Stopic* LM is used additionally, from 0.006 to 0.01. In terms of WER, no further improvement was achieved compared to the results presented in Table 6.6. For two-fold interpolation, a WER of 14.2% was achieved, for three-fold interpolation the WER was 14.1% (see Table 6.8). However, although the mixture coefficients are small, the WERs for the two-fold interpolations with *Stopic* or *Ltopic* are identical, showing that both corpora are relevant. Again, no correlation between the size of the *Ltopic* corpus and WER was observed.

(C) General presentation data: It was observed on Table 6.9 that sometimes the large speaker-dependent corpus, general in topic *Stopic* was more relevant in terms of perplexity than the more specific, topic-dependent corpora. Therefore, in this experiment it was investigated, if the relevance of the *Ltopic* corpora could be increased as well when merged into a single corpus.

After removing duplicate web pages, this corpus consists of 48,726,560 words. Again, the mixture coefficients for the two-fold and three-fold interpolation with *Stopic* were optimized on the *RT-06Sdev-dev* set, with *rt06Sbase* serving as a background LM. The results in Table 6.10 show a

	Ltopic		Stopic+Ltopic	
ID	PPL-i	WER-i	PPL-i	WER-i
t035	148	14.0%	145	14.6%
l003	148	14.2%	145	14.3%
t012	159	16.9%	155	16.6%
t032	138	11.9%	136	12.1%
t041	173	13.5%	175	13.2%
t042	167	11.3%	162	11.9%
Overall	154	13.9%	152	14.0%

Table 6.10: Results achieved with language model adaptation using web data collected from presentation slides on *lectDev*. In contrast to Table 6.8, the data collected for the different lectures was merged into a single corpus. The mixture coefficients were optimized on the *RT-06Sdev-dev* development set.

slight improvement in WER, compared to Table 6.8. Also, the perplexities differ slightly. Surprisingly, with two-fold interpolation a slightly better WER could be achieved, compared to with three-fold interpolation. This shows that relevant information for some lectures can also be found in the topic data collected for other lectures, and that the data collected for a lecture does not cover all the necessary information. Therefore, if enough lectures are available, merging all data into a single corpus to construct a general speaker-adapted language model may be a good idea, if the lectures are at least mildly related.

Conclusion

In this section, corpora specific to the presentation slide were collected using the proposed *tf-idf* based query generation technique. It could be shown that information extracted from the slides with this technique, data related in topic to the corresponding lectures could be retrieved.

For the three-fold interpolation consisting of the background language model and the *Stopic* and *Ltopic* LMs, the same WER was observed as when *Ltopic* is replaced with *Sseed*. A possible explanation for this is that the mixture coefficients for both mixture LMs are optimized on the same held-out set. This might be sub-optimal, because the held-out set differs in topic from the presentation.

Therefore, in the next section it will be investigated, if the mixture language model improves further as soon as data from the same speaker or with the same topic becomes available.

topic	lectOther	lectDev	lectEval
keynote		t035	
courseA	l005	l003	l043
talkA		t012	
courseB	t011, t033, t034	t032	
talkB	t039, t040	t041	t036, t037
talkC	t043	t042	t038, t044

Table 6.11: Classification of the *lecture* data into specific topic groups.

6.3.5 Using other Talks of the same Speaker

In this section, it is analyzed how additional data from the same speaker can be used. Depending on the type of transcripts available, manually transcribed or automatic, and if the data matches the topic or not, it was analyzed how the adaptation results are influenced. Therefore, the *lectOther* data as described in Section 4.2.4 was used for supervised (using manual transcripts) or unsupervised (using automatic transcripts) adaptation.

With the experiments described in the following, it should be investigated if data from the same speaker and topic can be used to improve the topic-dependent mixture language models. Therefore, the talks and lectures in *lectOther* have to first classified into specific topic groups, and, second, the groups were matched with the talks and lectures in *lectDev* and *lectEval*. In the first experiment (A), this classification will be used to optimize the mixture coefficients on the data of *lectOther* corresponding in topic, instead of using *RT-06Sdev-dev*.

The next experiments (B) and (C) are going a step further. For both the background language model will be re-optimized using the *lectOther* data instead of using the RT-06S held-out data. In the case of (B), the same corpora as allowed for *rt06Sbase* will used, and in the case of (C) all available corpora could be used. The goal of these experiments is to investigate, if even already adapted language models can be improved by the proposed topic adaptation scheme.

The last experiment (D) should prove the assumption that the WER reduces, if the mixture coefficients will be optimized topic-dependently for all language model components instead of keeping them fixed for the background language model.

Topic clustering and interpolation scheme: First of all, the talks and lectures were manually classified into specific topic groups. As can be seen in Table 6.11, three different topic groups are present in *lectOther*. These groups were then used for optimizing the mixture coefficients individually. For topics not present in *lectOther*, the coefficients of the language model components were optimized on the whole *lectOther* set. This means that for *l003* and *t042* the mixture weights were optimized on the corresponding *l005* and *t043*. For *t032*, the mixture coefficients obtained on *t011*, *t033*, and *t034* were averaged. For *t041*, the mixture coefficients obtained on *t039* and *t040* were averaged, and, for *t035* and *t012*, the mixture coefficients were optimized on the whole *lectOther* set. In all cases, three-fold interpolation between the background language model *rt06Sbase* and the *Stopic* and *Ltopic* components were used.

(A) Topic dependent mixture coefficients: Using the method described above, the overall WER was reduced by 3.4% from 14.5% to 14.0%. The detailed results are given in Table 6.12 in the section labeled *rt06Sbase*. The row labeled *S+Ltopic* shows the results after adaptation. Although the amount of data available per topic for optimizing the mixture coefficients is small, adapting the mixture model on each topic individually is better than adapting on the whole *lectOther* set by 0.2 points in WER. Moreover, by optimizing the mixture coefficients on the topics of *lectOther*, the WER was improved by 0.1 compared to optimizing on *RT-06Sdev-dev* (see Table 6.8). The most significant reduction in WER, by 5.9%, could be observed for *t042*, even though the talk on which the mixture coefficients were optimized has a duration of seven minutes.

Compared to the results presented in Table 6.10, where the topic data was merged into a single corpus and the mixture coefficients were optimized globally on *RT-06Sdev-dev*, the same WER was achieved. When performing the same experiment on *lectOther* but optimizing the mixture coefficients in a topic-dependent fashion, no further improvement was observed. Overall, no relation between the amount of topic data collected per speaker, or the amount of data used for optimizing the mixture coefficients, and the obtained WER was observed. In addition, the results show that with topic-specific data for optimizing the mixture coefficients, recognition accuracy improved.

(B) Speaker-dependent language model *lectBase*: By using the *lectOther* set for selecting the language model components and optimizing the mixture coefficients a new language model was built. This model will be referred to as *lectBase*. The mixture coefficients presented in Figure 6.2 show that in contrast to the *rt06Sbase* LM, the language model components built on *EPPS-S* and *AMI* transcripts and the *BN* text data are taken into account. Furthermore, the importance of the *UW-M* data is increased while for *SMNR* it is decreased. The perplexity on *lectDev* is 156. Table 6.12 shows the WERs achieved with this language model in the section labeled *lectBase*. It can be seen, that this new background language model outperforms the *rt06Sbase* language model on all talks and lectures.

When applying the topic dependent adaptation scheme as described above to this language model, it can be seen that the overall WER is re-

LM	t035	l003	t012	t032	t041	t042	Overall
Using manual transcripts							
rt06Sbase	14.7%	14.6%	17.4%	12.6%	14.3%	12.1%	14.5%
+ S+Ltopic	14.5%	14.1%	16.7%	12.2%	13.8%	11.2%	14.0%
lectBase	14.2%	14.5%	16.7%	12.1%	14.1%	11.4%	14.1%
+ S+Ltopic	13.8%	14.3%	16.2%	11.6%	13.5%	11.3%	13.7%
re-estimated	13.9%	14.0%	15.9%	11.5%	13.2%	10.8%	13.5%
lectAdapt	13.9%	13.9%	15.7%	11.5%	13.2%	11.1%	13.5%
+ S+Ltopic	14.0%	13.8%	15.7%	11.2%	13.3%	10.9%	13.4%
re-estimated	13.6%	13.9%	15.7%	11.2%	13.6%	10.7%	13.3%
Using automatic transcripts							
rt06Sbase							
+ S+Ltopic	13.9%	14.0%	15.9%	11.5%	13.2%	10.8%	14.0%
lectBase	14.3%	14.5%	16.8%	12.1%	14.0%	11.4%	14.1%
+ S+Ltopic	13.9%	14.3%	16.1%	11.6%	13.1%	10.7%	13.7%
re-estimated	14.1%	14.1%	16.1%	11.7%	13.4%	10.9%	13.7%
lectAdapt							
re-estimated	14.1%	13.7%	15.8%	11.4%	13.5%	11.1%	13.4%

Table 6.12: Results of different language model adaptation experiments by using speaker related data in the form of manual or automatic transcripts. The results are given for *lectDev*.

duced by 2.8%, even though *lectBase* is already adapted by the optimization of the mixture coefficients on the same data (*lectOther*). This clearly shows the relevance of the speaker- and topic-dependent data collected.

(C) Adapted speaker-dependent language model *lectAdapt*: The goal of this experiment was to see if it is possible to improve the performance of an already adapted language model with the topic-dependent adaptation scheme described above. Therefore, it was necessary to compute an already adapted language model by using the other corpora described in Section 4.2.3 and used for *LM6*. This language model is further referred to as *lectAdapt*. The mixture coefficients given in Figure 6.2 show that compared to *lectBase*, the components built on the web data collections *UKA-MP* and *UKA-LP* as well as on the proceedings *PROC* were added and the component built on *TED* was removed. In terms of WER, Table 6.12 shows an improvement of 4.3% to 13.5% compared to *lectBase*. Applying the topic-dependent adaptation scheme to this language model as a background model results in a slight decrease in WER by 0.1 points (*lectAdapt + S+Ltopic*).

(D) Global optimization: So far, the foreground language models were always interpolated with a fixed background language model. Therefore, it was analyzed if the WER improves when the mixture coefficients of the background language model were adjusted on the *lectOther* data as well. The mixture weights of the background language model and the foreground language models *Stopic* and *Ltopic* were optimized separately for each topic group. As can be seen in the rows of Table 6.12 labeled with *re-estimated*, the WER improved by 0.2 for *lectBase* and by 0.1 for *lectAdapt*. While for *lectBase* the WER for almost all talks and lectures was improved, for *lectAdapt* WER increases for *l003* and *t041*. Especially for *t041*, the mixture coefficients (which were averaged and optimized on *t039* and *t040*) show a huge difference compared to the coefficients optimized on *t041* itself. In particular, the *Ltopic* component should have had a coefficient of 0.21 instead of 0.05.

Conclusion

Overall, the proposed topic dependent adaptation scheme, described at the beginning of this section, can be used to adapt an unadapted language model (*lectBase*, WER 14.1%), such that an almost identical WER compared to a highly adapted language model is achieved, i.e. 13.5% (re-estimated *lectBase* vs. *lectAdapt*). Furthermore, even an already adapted language model (*lectAdapt*), built using huge amounts of collected web data, was improved from 13.5% to 13.3%. The best improvement was obtained, when the mixture coefficients for all language model components were re-estimated instead of using the background language model as a single component.

6.3.6 Using Automatic Transcripts

The most interesting results were achieved when the automatically transcribed data was used instead of the manually transcribed data. The interpolation weights for *rt06Sbase* were kept unchanged, but for *lectBase* and *lectAdapt* the mixture coefficients were re-estimated on ASR hypothesis of *lectOther*. As can be seen from the results given in Table 6.12, after *S+Ltopic* is applied to the background language models, the WER improved, but not by as much as when manually transcribed data is used. The difference amounts to 0.2 for the *re-estimated S+Ltopic lectBase* LM, and to 0.1 if *lectAdapt* is used as a background LM.

Conclusion

The results in this Section show that automatic transcriptions from recorded talks and lectures from past uses of the lecture translation system can be easily used for adaptation. At an automatic transcription word error rate of less than 15%, a time consuming manual transcription of the recordings is no longer necessary.

6.4 Expanding the Vocabulary

For an efficient decoding process in a speech recognizer, a fixed vocabulary is indispensable, because only then can the search network be precomputed and optimized for decoding speed. In addition, an increase in the vocabulary size always involves an increase in decoding speed due to the larger search space, as well as in an increase in memory requirements due to the increased number of transitions in the n-gram language model. On the other hand, with a fixed vocabulary, out-of-vocabulary (OOV) words cannot be recognized, and each occurrence of an OOV word is in average responsible for 1.5–2 recognition errors [GFS98].

Generally, the recognizer's vocabulary is selected in order to obtain an OOV-Rate which is as low as possible on the target task or domain by minimizing the OOV-Rate on a related development set [IO99, VW03]. Technical talks and lectures have a narrow vocabulary, but the vocabulary can differ largely from talk to talk due to the technical terms and expressions. In addition, correct recognition of especially these technical terms and expressions is very important for the understanding of the context of the talk. The same is true for proper names which are often missing in the recognizer's vocabulary.

As in [RS02], a method was developed to decrease the OOV-Rate based on presentation slides and on the *Sseed* corpus. In order to prevent misspellings, the selection of words per topic has to be performed semi-automatically. The developed heuristic is based on the assumption that the longer the word, the more meaningful and relevant it is:

1. Select all non-singleton words from a given corpus which do not exist in the baseline vocabulary.

2. Remove all words shorter than three characters.

3. Check all words from four up to six characters manually, and remove all meaningless words.

4. All words left will be used for vocabulary adaptation.

In order to use the additional words in the speech recognizer, pronunciations have to be generated. Therefore, the pronunciation generator developed for the RT-06S evaluation [FIK+06] was used, which first tries to find the word in a large background pronunciation dictionary, and, if not found, uses the Festival [BT97] speech synthesis system for pronunciation generation.

We pose the question of how to add the selected words to the language model. If the frequency of the words in the given corpus is low, using this corpus as is for language modeling might underestimate their transition probabilities. While for the large *Stopic* corpus this might not be a problem,

talk	OOV	WER	OOV	WER	OOV	WER	OOV	WER
	baseline		general		specific		both	
t035	0.62	13.9%	0.32	13.5%	0.34	13.6%	0.30	13.6%
l003	0.45	14.0%	0.28	13.6%	0.25	13.7%	0.22	13.6%
t012	0.45	15.9%	0.27	16.0%	0.45	15.8%	0.27	15.8%
t032	0.37	11.5%	0.30	11.3%	0.37	11.2%	0.30	11.2%
t041	1.08	13.2%	0.97	13.0%	0.97	12.9%	0.92	12.9%
t042	0.62	10.8%	0.44	10.9%	0.62	10.9%	0.44	10.9%
Overall	0.49	13.5%	0.33	13.3%	0.38	13.2%	0.30	13.2%
lectAdapt								13.0%

Table 6.13: Comparison of OOV-Rates and WERs for different expansions of the vocabulary on *lectDev*. As baseline language model, the *re-estimated S+Ltopic lectBase* language model was used for the detailed results. The result in the last row was achieved with a similar adapted *lectAdapt* LM. WERs were obtained after performing language model rescoring.

using the presentation slides for language modeling might not be a good idea. Furthermore, it cannot be guaranteed that the selected words occur frequently enough in the collected *Ltopic* corpus. Therefore, all selected words were added to a special class in the language model, i.e. the unknown word class, to which all words not in the vocabulary are mapped during language modeling. The unigram probability of the unknown word class for large corpora is typically comparable to the probability of 'a', which is rather high. An uniform distribution is used as the intra-class distribution for the selected words, because the frequency of the selected words in the corpus does not match real usage.

As shown by Schaaf [Sch04] by adding the new words to a single unknown word class as described above, relatively good recognition accuracy could be achieved. However, when adding a large number of words, this method may be disadvantageous because the words are indistinguishable from a language model point of view. Schaaf investigated also more sophisticated methods in which several unknown word classes were incorporated in the language model. Words are then added to one or multiple classes. Therefore, the experiments described in the following should be seen as a first prove of the above described selection procedure. It can be expected that improvements observed by adding the selected words to a single unknown word class carry over when multiple classes are used.

6.4.1 Experiments

Table 6.13 shows the WERs together with the OOV-Rates achieved with different expansions of the vocabulary. The *re-estimated S+Ltopic lectBase* language model was used as the baseline language model for the results in the top part of the table. As can be seen, the baseline vocabulary achieved an overall OOV-Rate of 0.49 and an overall WER of 13.5%. In the following three experiments it will be shown that the selected words improve the recognition results. In the first experiment, words were selected from the *Stopic* corpus and in the second experiment from the presentation slides. In the last experiment both selections were combined.

Extracting words from general speaker related data

The above described heuristic selected 715 words from the *Stopic* corpus. As can be seen in the columns labeled as *general* in Table 6.13, the overall OOV-Rate was reduced by 33% to 0.33 and the overall WER to 13.3%. From the detailed results, it can be seen that not all talks or lectures benefit from the selected words. For *t012* and *t044* the WER increases slightly although the OOV-Rate decreased significantly. The reason for this is the above described problem when adding too many words to a single class using an uniform intra-class probability distribution.

Extracting words from the presentation slides

In this experiment, the heuristic was used to select words from the presentation slides. The number of words selected, ranged from 2 to 182, with an average of 40. Although the OOV-Rate is higher than compared to the first experiment, the overall WER could be reduced to 13.2% (column labeled as *specific* in Table 6.13). The reasons for this difference to the first experiment are that first, the amount of words selected is smaller and therefore better distinguishable, and, second, the words selected are more relevant.

The combination of both selections

In the last experiment, both selections were combined. As can be seen from the results in the column labeled as *both* in Table 6.13 the overall OOV-Rate was further reduced to 0.30 but the overall WER remained constant.

Conclusion

Overall, the developed vocabulary selection heuristic was able to reduce the OOV-Rate by 39% and the WER by 2.2% relative. But, as already expected, adding all selected words to a single class is unsuitably when the number of words selected is large. If the *re-estimated S+Ltopic lectAdapt* language model was used instead of the *re-estimated S+Ltopic lectBase* language model, the overall WER was reduced by 2.3% relative to 13.0%.

6.4.2 Handling of foreign words

If a speech recognizer is developed for a single language, the phoneme set typically covers pronunciations restricted to that language only. This makes it difficult to model foreign words coming from other languages such as proper names, because the necessary phonemes do not exist in the recognizer. As an example, it is difficult to model German umlauts such as in the thesis author's last name "Fügen" with an English recognizer. For the recognition of technical talks or lectures in other languages, this is even more a problem because many technical terms exist in English only due to the dominance of English for international communication. Moreover, the pronunciation of "Fügen" depends on the mother tongue of the speaker and on the speaker's experience with foreign languages. An English native speaker might pronounce "Fügen" differently from a German native speaker giving a talk in English.

The key to a solution is the use of a multilingual acoustic model [SK06] instead of a monolingual one. This makes it possible to model foreign words in other languages with the corresponding phoneme sequence. Moreover, further pronunciation variants can be added with phonemes of other languages to support non-native pronunciations of the same word. A disadvantage is that the recognizer increases in size and complexity. Furthermore, depending on the modeling technique, the introduced pronunciation variants may cause the recognition accuracy to suffer. Modeling techniques can range from completely language-independent acoustic models, over cross-language acoustic models [SW01], up to cross-language acoustic and language models [FSS+03, HHP01]. With the latter technique, it is possible to use the same recognizer for two or more languages.

For the developed prototype of a simultaneous translation system, multilingual acoustic models were not used because the expected performance gain was small on the given data set.

6.5 Topic Adaptation for Machine Translation

Adaptation of the MT component of the baseline translation system towards the more conversational style of lectures and known topics was accomplished by a higher weighting of the available lecture data in two different ways. First, for computing the translation models, the small lecture corpora were duplicated several times and added to the original EPPS training data. This yielded a small increase in MT scores (see Figure 6.15).

Second, for (target) language model adaptation, web data was collected. As can be seen in Table 6.14, the baseline *EPPS-T* language model based on EPPS text data only has a very high perplexity. Therefore, two different web data collections were performed. For further adaptation of the language model to the more conversational style of lectures, common spontaneous

	lectDev	lectEval	weight
EPPS-T	631	1019	0.9%
WEB01, common speech web data	342	554	6.7%
WEB02, topic related web data	217	544	57.2%
lectures		711	35.1%
Interpolated		348	

Table 6.14: Comparison of the perplexities for differently adapted target language models.

speech expressions were extracted out of some speech transcripts following the approach of [BOS03], and used as queries. The resulting corpus was perplexity filtered to increase its impact. The resulting corpus had size of about 90M words and is referred to as *WEB01*.

A second collection was performed in the same manner as above by using topic related n-grams as queries, where the n-grams were extracted out of the development data using *tf-idf*. This results in a corpus which we will refer to as *WEB02* of about 45M words. As can be seen from Table 6.14, both corpora yield significantly lower perplexities than *EPPS-T*. Note that due to the lack of additional data for system development to the time when these experiments were made, the query extraction was performed on *lectDev*. As a result, the perplexities given in Table 6.14 for *WEB02* on *lectDev* are biased.

In a third step, additional lecture data (*lectDev*) was added. On *lectEval*, this data achieved a perplexity of 711. From the interpolation weights given in Table 6.14, which were tuned on other held-out data, it can be observed that the relevance of the *WEB02* data is much higher than that of the *WEB01* data. This shows, that by using the developed topic adaptation technique more relevant data is retrieved than when using queries which cover spontaneous speech in general. Overall a perplexity of 348 on the evaluation data was achieved.

The effects of translation and language model adaptation and their combination, are shown in Table 6.15 for English-to-Spanish. In absolute terms, the translation performance on this difficult task is still quite poor when compared with tasks for which large amounts of training data, similar in style, are available, such as the TC-STAR EPPS task. Nevertheless, small amounts of lecture data were sufficient to significantly improve performance, especially when amplified by using language model adaptation with similar web data.

	Unadapted		TM-adapt		+LM-adapt	
	NIST	BLEU	NIST	BLEU	NIST	BLEU
t036+, text input	5.72	23.4	5.89	25.0	6.33	28.4
t036+, ASR input	5.06	17.9	5.15	18.9	5.60	23.0
l043, text input	5.27	19.6	5.35	20.3	5.48	21.6
l043, ASR input	4.80	16.6	4.85	16.9	5.05	19.0

Table 6.15: Translation model and language model adaptation. *t036+* is a merge of *t036* and *t037*.

6.6 Conclusion

In this chapter, an adaptation framework for lectures and speeches was proposed. Depending on the amount of information available for a particular speaker or topic, different adaptation levels can be used within a simultaneous translation system. While the previous chapter's focus was acoustic model-based speaker adaptation, this chapter's main focus was topic adaptation. The proposed technique is the collection of additional speaker-related or on-topic data which is used for linear interpolation with a background language model. The data is collected through the Internet by querying a search engine and retrieving and filtering the resulting web pages. For query extraction, a *tf-idf*-based heuristic was developed, which computes *tf-idf* weights for n-grams instead of single words and uses these weights together with frequency-based thresholds. The n-grams were extracted from a seed corpus, i.e. either manually collected speaker information such as research papers (*Sseed*) or presentation slides (*Lseed*), and each of the extracted n-grams was used as a single query. The main problem is the optimization of the mixture coefficients for the mixture language model. The better the data match in speaking style and topic, the better can the resulting mixture language model be expected to perform. Therefore, several different data sets were tested for suitability, from data matching in speaking style to data matching also in topic. Adaptation by using the reference transcripts and adaptation by using the speech recognizer's hypotheses was compared. Significant improvements were observed on *RT-06SDev* as well as on *lectDev*.

Using speaker-related information only, three-fold language model interpolation improved the word error rate by 3.8% on *lectEval*. For interpolation, language models built from the seed corpus (*Sseed*), the corpus collected from n-grams extracted from the seed corpus (*Stopic*), and an unadapted background language model (*rt06Sbase*) was used. The mixture coefficients were optimized globally on some held-out data.

When other talks or lectures of the speaker were available, this data (*lectOther*) was used for further language model adaptation. First, the unadapted background language model was optimized on *lectOther*, leading

	lectDev	lectEval
rt06Sbase	14.5%	15.8%
+ Sseed+Stopic	14.1%	15.2%
+ S+Ltopic (ref)	14.1%	15.1%
+ S+Ltopic (hyp)	14.0%	15.0%
lectBase	14.1%	15.4%
+ S+Ltopic re-est. (ref)	13.5%	14.8%
+ new words	13.2%	14.8%
+ S+Ltopic re-est. (hyp)	13.7%	14.6%
lectAdapt	13.5%	14.5%
+ S+Ltopic re-est. (ref)	13.3%	14.4%
+ new words	13.0%	14.4%
+ S+Ltopic re-est. (hyp)	13.4%	14.4%

Table 6.16: Final topic adaptation results in WER on *lectDev* and *lectEval*.

to a WER reduction of 2.5% to 15.4% on *lectEval*. Second, the reference transcripts of these talks or lectures were used as seed data for collecting other topic-related data (*Ltopic*), as well as for optimizing the mixture coefficients topic-dependently. This resulted in a further WER reduction of 3.9% to 14.8%. When using ASR hypotheses of the lecture or talks, the results were mixed. On *lectDev*, the WER was higher than that achieved with reference transcripts, but on *lectEval* the WER is lower.

Furthermore, it was shown that an already adapted language model (*lectAdapt*), consisting of several language model components for which the interpolation was optimized on *lectOther*, can be improved with the proposed topic-dependent adaptation scheme. On *lectEval*, the WER achieved with *lectAdapt* was improved from 14.5% to 14.4%. As for *lectOther*, the improvements achieved when using ASR hypotheses compared to the use of reference transcripts for between *lectDev* and *lectEval*. On *lectEval*, the same WERs were obtained.

In summary, the results show that with the developed topic-dependent adaptation scheme as well as the *tf-idf* based adaptation procedure, a general background language model can be significantly improved. The most important result is that in our case ASR hypotheses are as suitable as reference transcripts for optimizing the mixture coefficients, which allows for fully automatic processing of the adaptation framework.

In addition to language model adaptation, the vocabulary was extended with technical expressions or proper names found in the presentation slides and the speaker-related *Sseed* data. For extracting the relevant OOV words, a heuristic was developed which successfully reduced the OOV-Rate on *lectEval* from 0.77 to 0.50. On *lectDev*, the WER was improved significantly, but on *lectEval* no improvement was observed. An explanation for this may be the use of a single class in the language model to which all new words are added. More classes, as proposed in [Sch04, RS02], may need to be used.

The *tf-idf*-based query generation and web-data collection technique was also successfully used for target language model adaptation in the machine translation component. With four-fold interpolation consisting of the European Parliament Plenary Session-based background language model, some Spanish lectures from *lectDev* and two language models build from web data, the perplexity on *lectEval* was reduced from 1019 to 348. One web corpus was collected with common spontaneous speech expressions, the other one by using the proposed *tf-idf*-based topic collection technique. As a result of the perplexity reduction, the translation quality on ASR hypotheses of a system in which the translation model is already adapted was further improved significantly by 21.7% to 23.0% for *t036+* and 12.4% for *l043*.

Chapter 7

Latency and Real-Time

As already described in the introductory chapters, simultaneous interpretation is in contrast to consecutive interpretation defined by an interpretation of the source speech as quickly as possible. Hereby, the length of the ear-voice-span or latency depends on the grammatical structure of the source and target language, but should be as short as possible. Therefore, the whole system requires high processing speed of the components and a short latency. On the other hand, a good recognition and translation quality is required to facilitate a fluent translation. But speed and quality are two system attributes, which are orthogonal to each other — increasing the speed of a system often reduces its quality and vice versa.

This chapter deals with the question on how and to what extent the speed and latency of the system can be increased without loosing to much recognition and translation quality. To increase the processing speed, tuning can be done with respect to single components; however, shortening the latency requires adjustments to the whole framework. To shorten the latency, it is necessary to de-serialize the processing, in which the ASR starts to decode when the recording is finished and transmits the hypothesis to the MT not until the decoding has been finished, to an overlapped processing. Further speed-up can be achieved if the ASR component transmits partial hypothesis while decoding the current recording as shown in Figure 7.1. This allows the MT component to start earlier.

To support this kind of processing, incremental adaptation and the language model have to be adjusted in the ASR component; the Ibis decoder used in this thesis already processes the data time-synchronously, i.e. frame-by-frame. On the other hand, the non-linearity of the translation process, due to possible re-orderings of words or phrases makes the same changes in the MT component much more complicated. Therefore, almost all current machine translation systems require sentence-like units as input which are then translated as a whole. For this reason a *Resegmentation* component was introduced, which is responsible for chunking the continuous output

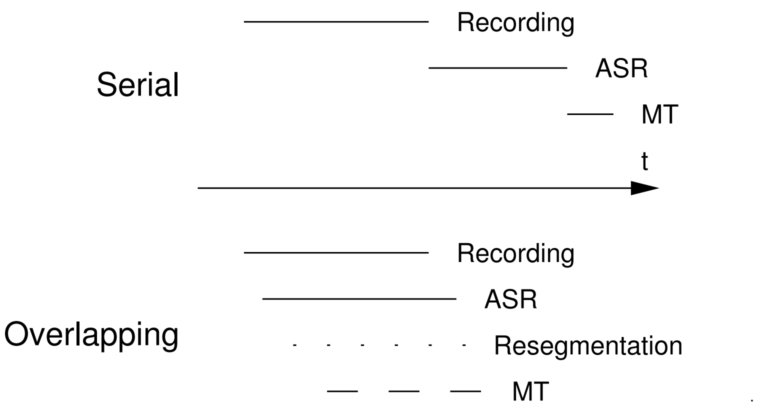


Figure 7.1: Comparison of serial and overlapping processing.

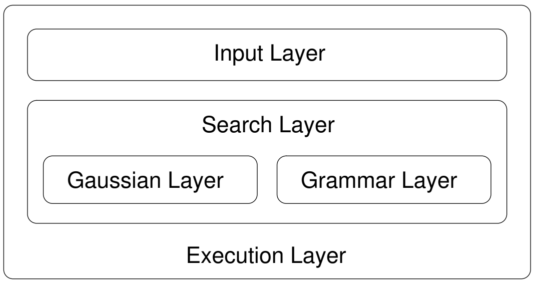
stream of the speech recognizer into smaller segments for which the translation quality does not suffer.

The remainder of this chapter focuses on the real-time and latency aspects of a simultaneous translation system. The *Resegmentation* component will be discussed in Chapter 8. To increase the real-time of a speech recognizer, search-space pruning and Gaussian selection techniques are analyzed in Section 7.2 and 7.3 using the Ibis decoder. In Section 7.4, methods are introduced to reduce the latency of the serialization, and their influence on incremental adaptation and recognition quality are analyzed. Section 7.5 concludes this chapter.

For the following experiments, several different acoustic models were trained on the same training data as used for the baseline system, which was described in Section 4.2.1. They differ by the overall number of codebooks ($m \in 3000, 4000, 6000$) and the maximum number of Gaussians per codebook ($n \in 64, 128, 256$) and are named with $m - n$ for an acoustic model with m codebooks and a maximum number of n Gaussians per codebook. If not otherwise specified, all acoustic models were trained up to step 5 of the training procedure described in Section 4.2.1. Acoustic model specifiers tagged with *MMIE* are trained up to step 9. The acoustic model used for the baseline system described in Section 4.3.7 is referred to as *4000-64-mmie*. Table 7.1 shows the total number of Gaussians available in each of the acoustic models together with its word error rate and real-time factor. If not otherwise stated, the real-time factor of the decoding process itself is given, without the time necessary for incremental adapting the cMLLR and VTLN parameters, and is referred to as *RTF* or *RTF decoding*. *RTF total* refers to the total amount including adaptation.

acoustic model	number of Gaussians	WER	RTF
3000-64	184254	15.6%	1.34
3000-128	328623	15.3%	1.71
3000-256	497728	14.9%	2.18
4000-64	232046	15.5%	1.25
4000-128	390774	15.1%	1.77
4000-256	539345	14.9%	2.13
6000-64	316917	15.4%	1.34

Table 7.1: Overview of the total umber of Gaussians for each acoustic model together with its word error rate on *lectDev*.



Layer	Examples
Input Layer	Frame-skipping, down-sampling
Search Layer	Beam search, search-space pruning
Gaussian Layer	Gaussian selection
Grammar Layer	Language model lookahead
Execution Layer	SIMD, compiler optimizations

Figure 7.2: A five layer model for the categorization of optimization strategies of a speech recognizer.

7.1 Speed-up Techniques for Speech Recognition

Following [Kö04, CSMR04], a five-layer model for the categorization of optimization strategies of a speech recognizer is introduced and shown in Figure 7.2. The motivation behind this model is that single optimizations can be categorized into one level and that optimizations on different levels have more likely a greater impact in reducing the decoding speed, than compared to different optimizations on the same level, because they are independent to each other. In contrast to [Kö04, CSMR04] the presented model categorizes the whole decoder into 5 different layers instead of categorizing the Gaussian mixture model computation only.

The *Execution Layer* covers all techniques related to the execution of the decoding process. This includes manual optimizations such as the use of SIMD (single input, multiple data) instructions for matrix operations and observation probability estimation [KSN00], as well as automatic optimizations made by modern compilers. In this thesis, for compiling Ibis, the following settings were used in combination with Intel's C++ Compiler 10.1 for Linux on a 64bit CPU: `-axW -O3 -ip -parallel`. This generates specialized code paths for SSE2 and SSE instructions for Intel processors (`-axW`), turns on aggressive optimizations such as loop and memory access transformation, turns on pre-fetching (`-O3`), enables additional intra-file inter-procedural optimizations (`-ip`), and uses the auto-parallelizer to generate multi-threaded code for loops that can be safely executed in parallel (`-parallel`) [Int08]. Compared to no optimization at all, the resulting speed-up can range from 10%-20%. In another experiment, a manual optimization of the observation probability computation by using SIMD instructions similar to [KSN00] was compared to the automatic optimizations of the Intel compiler, and no difference in performance could be measured between the two.

Techniques belonging to the *Input Layer* are responsible for reducing redundancies in the input stream, i.e. the sequence of observation vectors. Well-known techniques are the so-called frame-skipping or down-sampling. While down-sampling [CSMR04] reduces the amount of data in the input stream by just taking every n-th observation vector, frame-skipping is more intelligent. Only those observation vectors are ignored for which the distance to the preceding vector is smaller than a given threshold [Wos98, KFSW05]. Another common approach is to use a voice-activity detector to remove observation vectors which can be classified as non-speech.

In the *Search Layer*, all techniques are concentrated which focus on optimizing the decoding process itself as well as the architecture used for representing the search space. Search-space pruning is one technique which can be categorized in this layer, architectural optimizations such as pronunciation prefix trees together with the linguistic polymorphism used in Ibis also belongs to it. Both will be explained below. Other architectures use weighted finite-state transducers (WFST), which allow to efficiently encode all the various knowledge sources. The network resulting from the composition of these WFSTs, after minimization, can be directly used in a Viterbi decoder [MPR00, SPZ05]; this approach has been shown to be quite promising [KNRM02].

Closely related to the *Search Layer* are optimizations categorized in the *Gaussian* and *Grammar Layer*. Two widely used techniques for speeding up the observation probability computation in the *Gaussian Layer*, the BBI algorithm and Gaussian Clustering, will be discussed below. For fast access to the language model probabilities, the use of a language model lookahead tree is very common [ONE96]. Pruning and smoothing techniques are also used for language models, but are more related to reducing the size of the language models instead of accelerating the probability computation. In

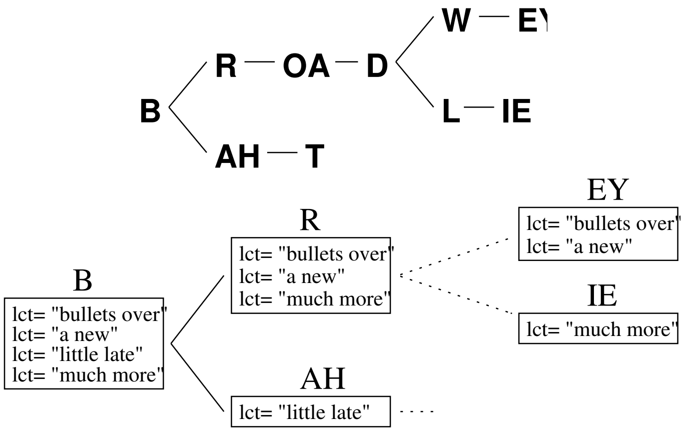


Figure 7.3: A simple pronunciation prefix tree without and with linguistic morphed instances for a system using context independent acoustic models. “lct” stands for linguistic context.

fact, for the commonly used n-gram back-off language models, pruning can slow down the probability computation, because the backing-off to lower n-grams needs additional memory accesses.

Independent of algorithmic improvements are the quality of the acoustic and language models. The effectiveness of several of the above mentioned techniques depends on the similarity between training and testing conditions.

7.2 Search-Space Pruning in Ibis

The search-space of the decoder is restricted by the number of pronunciations in the dictionary and the size of the acoustic model. An elegant way of representing the search-space of a speech decoder is by using pronunciation prefix trees (PPTs). A pronunciation prefix tree is generated out of the pronunciation dictionary by merging identical phone sequences of different dictionary entries and convolving them with the acoustic model. A simple example for context independent acoustic models is given in Figure 7.3. For further compactification of the search-space, Ibis allows compressing the PPT into a search network by removing other redundancies between different branches with an iterative algorithm.

When reaching the end of a PPT, i.e. its leafs, it becomes necessary to extend the search to the following word candidates. This can be done either by creating a copy of the tree or by re-entering the tree and keeping track of the word history. The idea of the tree copy process [NHUTO92, Ode95, OEN98, Aub99] is to create a separate copy for each surviving linguistic state[1] after path recombination. Since the number of different linguistic states, i.e. tree copies, can be a few hundred for a long-span language model, efficient pruning is necessary. Ibis uses an alternative approach to the tree copying process, which is based on a single PPT. Therefore, a linguistic polymorphism as described in [SMFW01, SMFW02] and similar to [AHH96, FKFW99] was introduced. In each node of the PPT, a list of linguistically morphed instances is kept, and each instance stores it's own back-pointer and score for the underlying Hidden Markov Model (HMM) with respect to the linguistic state of this instance. Since the linguistic state is known, the complete language model information can be applied in each node of a PPT. The advantage of this search-space organization is that efficient pruning can be applied, which eliminates the sub-tree dominance problem of the tree copy approach.

Pruning within the search network can now be performed at different levels and by using different techniques. To clarify the experiments in the following sections, the levels will be described shortly. In the description, the values in parenthesis are the default pruning parameters. Ibis is a time-synchronous decoder, which means that in each frame all active decoding paths are expanded and pruned before entering the next frame, and all active decoding paths end at the current frame. A decoding path is active, if it will be expanded in the next frame, i.e. was not not pruned. Pruning can be either done relative to the current best, or by taking the top N best out of a sorted set of paths.

State Level: All states of the underlying HMM structure belong to the *state level*. Pruning at the state level means that transitions from one state to another state are done only, if the transition score is not exceeding a specific threshold. This threshold is defined relative to the score of the best state transition in the current frame. Pruning is performed after all transition scores for the current frame are computed. (*stateBeam=130*)

Linguistic Instance Level: All linguistic morphed instances, i.e. all paths which are in the same HMM state for the current frame, but have different linguistic contexts, belong to the *linguistic instance level*. All instances with the same HMM state are combined in a single PPT node. Especially, if the path comes to an word-end, i.e. PPT leaf, it is important to keep the numbers of instances per state as small as possible to reduce the computational effort due to the fan-out of the right context models for starting the new words. Therefore, new instances will not be added if their score exceeds a threshold relative to the best score in a PPT node. Furthermore, the numbers of instances per node

[1] The current word represented by the PPT leaf and it's predecessor words are referred to as the linguistic state.

are pruned to a fixed number using topN pruning. (*morphBeam=80, morphN=10*)

Word Level: Transitions from the last state of one word to the next state of another word, is not allowed, if the score exceeds a threshold relative to the best score of the current frame. Furthermore, the number of these transitions is reduced with the help of a topN pruning. In contrast to the pruning at the linguistic instance level, which is done per node only, the pruning at the word level limits the overall number of words which can be followed in the next frame. (*wordBeam=90, transN=40*)

A so called *master beam* (default *mb=1.0*) can be used to adjust all relative pruning parameters by a given factor.

7.2.1 Experiments and Discussion

The goal of the following experiments is to show the relationship of different pruning and language model parameters and their influence on decoding speed and error rate. Since the Ibis decoder is a single-pass decoder, another goal was to evaluate the possible improvement due to a second language model rescoring pass. As a baseline, the decoding performance using default pruning (see above) and language model parameters was used. Specifically, a language model weight of 26 and a word penalty of 6 without any language model rescoring was used.

First, the relationship of all relative pruning parameters was analyzed. All language model and topN pruning parameters were kept at the default values. From Figure 7.4, it can be observed that the *state beam* has the largest influence on recognition accuracy. Changes in the *word beam* only slightly influence WER down to a value of around 70-80, but values below this break-point result in a rapid increase of WER. The same is true for the *morph beam*, where the break-point is around 50-60. In terms of decoding speed, it can be seen that there is a almost linear correlation between all relative pruning parameters and the real-time factor, besides *morph beam*, which does not influence the real-time factor at values larger than 50.

Figure 7.5 concentrates on the relationship between all topN pruning parameters with respect to the WER and RTF. All language model and relative pruning parameters were kept fixed. Regarding *transN*, it can be seen that there is almost no change in WER down to a value of 12. Below this break-point WER begins to increase. For *morphN*, the break-point in WER is at 3. Decoding speed increases continuously with decreasing *transN* and *morphN*, and the influence is larger for *transN* than for *morphN* especially for small *transN*. This is clear because the smaller the allowed number of transitions, the smaller the number of possible linguistically morphed instances. Given these results, by tuning the topN pruning parameters the

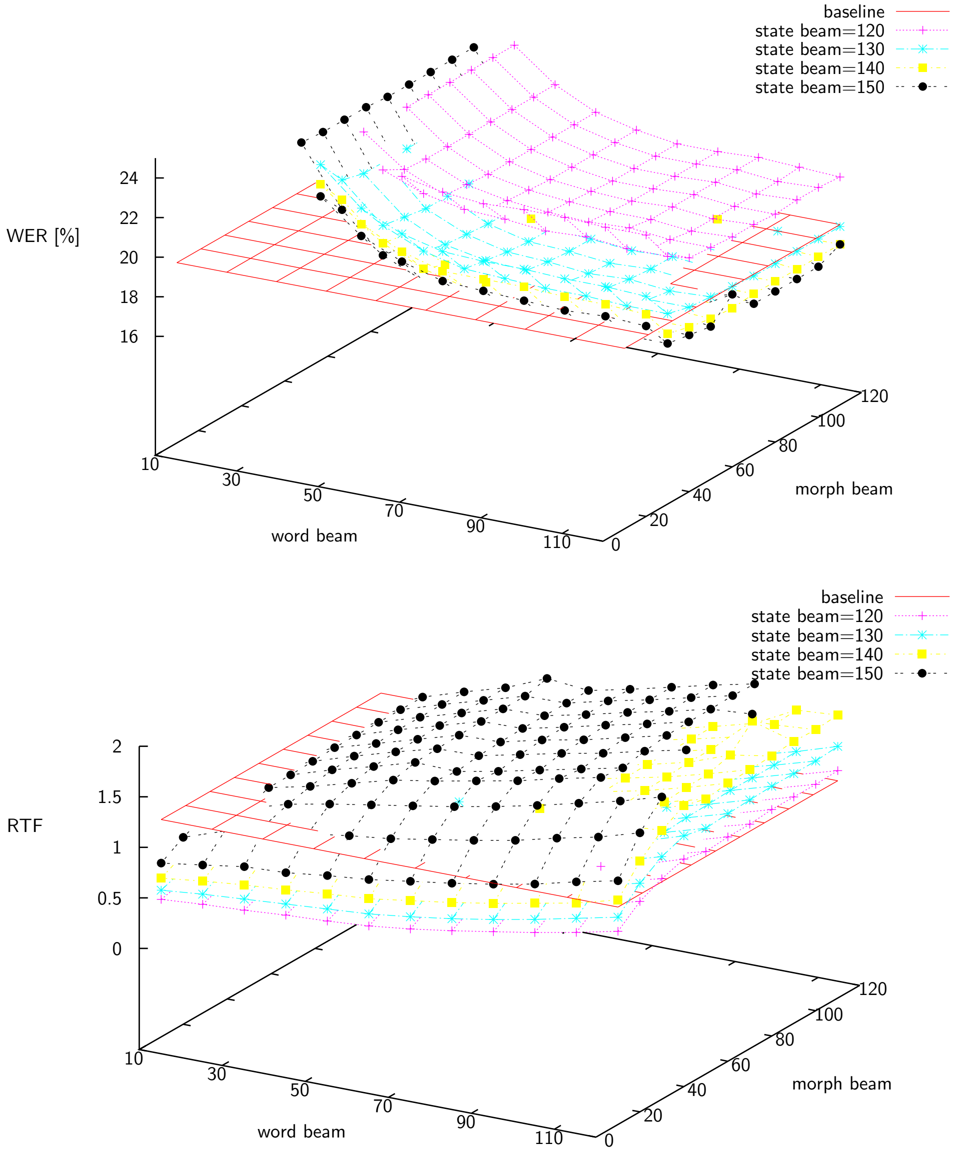


Figure 7.4: The relationship of different relative pruning parameter settings to word error rate (WER) and decoding speed (RTF) in Ibis. For the baseline, decoding was done with default pruning and language model parameter settings.

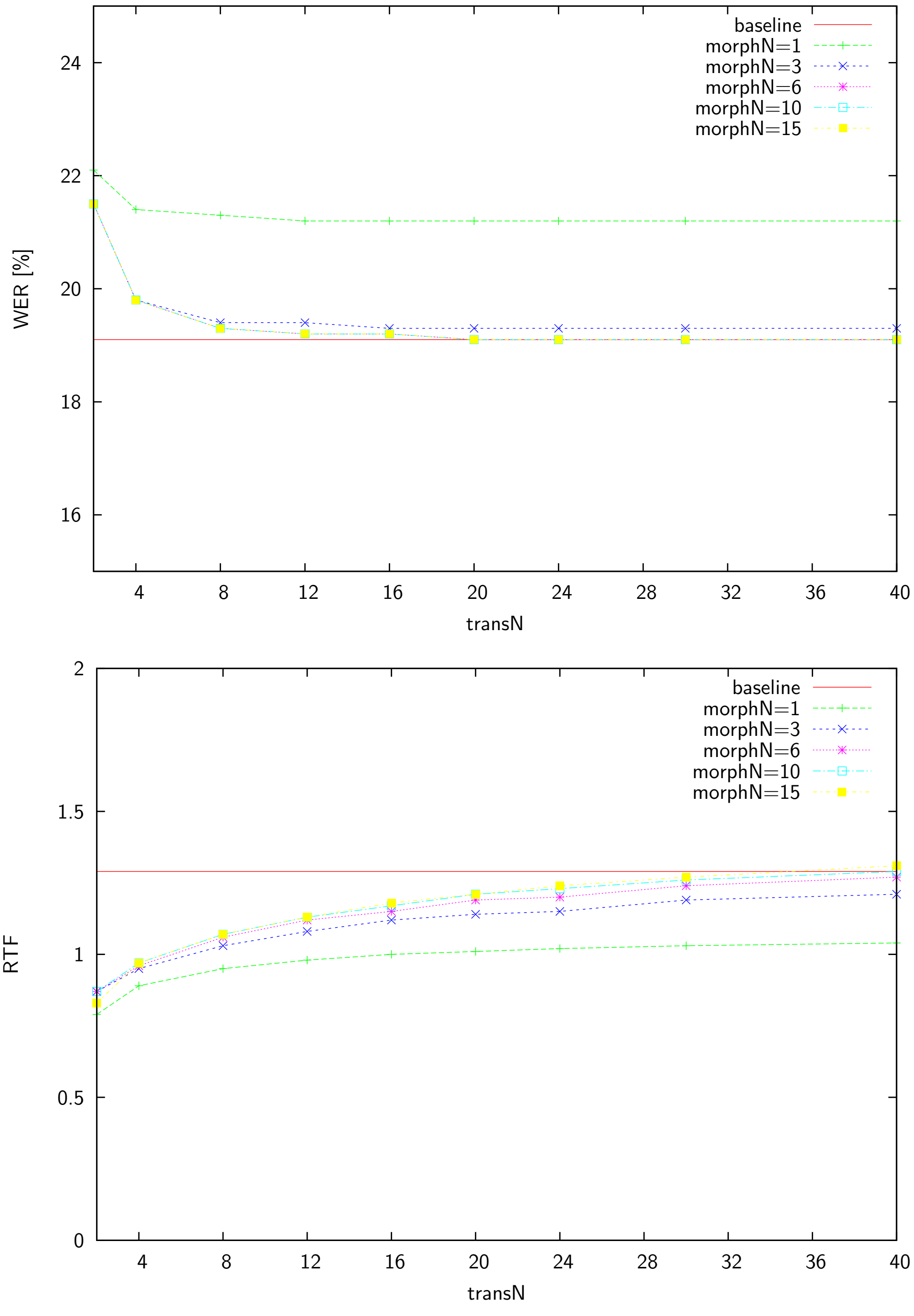


Figure 7.5: The relationship of different topN pruning parameter settings to word error rate (WER) and decoding speed (RTF) in Ibis. For the baseline, decoding was done with default pruning and language model parameter settings.

		WER	RTF
baseline	4000-64-mmie	14.3%	1.07
	4000-128-mmie	14.4%	1.57
	3000-64-mmie	14.4%	1.16
	3000-256-mmie	14.6%	2.01
optimized	4000-64-mmie	14.6%	0.89
	4000-128-mmie	14.6%	1.22
	3000-64-mmie	14.6%	0.88
	3000-256-mmie	14.7%	1.51

Table 7.2: Comparison of recognition accuracy and decoding speed between different acoustic models using default or optimized pruning parameters for decoding on *lectDev*. For the optimized results a *state beam* of 130, *word beam* of 80, *morph beam* of 60, *transN* of 20, and a *morphN* of 6 was used.

decoding speed can be increased without significant increase in WER. A *transN* of 20 and a *morphN* of 6 seem to be optimal.

Altogether, with the improved pruning parameter settings, the real-time factor could be improved by 17% to 0.89 for the baseline system (*4000-64-mmie*) with an increase in WER by 2.1% to 14.6%. The optimized pruning parameters also hold for systems using other acoustic model sizes, as can be seen in Table 7.2. The improvement for the *3000-64-mmie* is even larger; the RTF is reduced by 24% while the WER increases by only 1.4%.

Language Model Rescoring

In a single pass decoder, all knowledge sources are typically integrated as early as possible into the score computation. An example for this is the language model score, for which the exact score is applied as soon as the word identity is known. This can be already before a leaf node of the pronunciation prefix tree is reached. Prior to this, the best language model score given all possible words through the current state is applied. The following experiments intend to show whether a language model rescoring using the word lattice produced during decoding can still improve performance, and if the same improvement with respect to WER and RTF is possible using different language model and pruning parameters already during decoding, i.e. without the necessity of a rescoring.

For speech recognition, we seek to find the sequence of words $\hat{w}$ that maximizes the posterior probability for a given sequence of acoustic observation vectors $\mathbf{o}$

$$\begin{aligned}\hat{w} &= \underset{w}{\operatorname{argmax}}\, P(w|\mathbf{o}) \\ &= \underset{w}{\operatorname{argmax}}\, \frac{p(\mathbf{o}|w) \cdot P(w)}{p(\mathbf{o})} \\ &= \underset{w}{\operatorname{argmax}}\, p(\mathbf{o}|w) \cdot P(w)\end{aligned}$$

where $p(\mathbf{o}|w)$ is modeled by the acoustic model and $P(w)$ by the language model. Due to different model spaces, the variances and means of the acoustic and language model probabilities heavily deviate from each other. As a result of this a language model weight has to be introduced. Additionally, a word penalty has to be introduced, which depends on the length of the current hypothesis to prefer the recognition of longer words over shorter ones [Rog97]. With these additional parameters and after transforming all probabilities into the log-domain, the score $\log P(w|\mathbf{o})$ is computed as follows

$$\begin{aligned}\log P(w|\mathbf{o}) &= \log(p(\mathbf{o}|w) \cdot P(w)^{lz} \cdot q^{|w|}) \\ &= \log p(\mathbf{o}|w) + lz \log P(w) + |w| \log q \\ &= \log p(\mathbf{o}|w) + lz \log P(w) + |w| lp\end{aligned}$$

with $|w|$ being the length of the word sequence w, lz the language model weight, and lp the word penalty.

Given the explanations above, it follows that, given a fixed acoustic and language model, changes in lz linearly scale the overall score, while changes in lp cause an additive shift of the scores depending on the length of the hypothesis. For single-pass decoding with Ibis, this means that to achieve the same WER after scaling lz and lp, further referred to as *language model parameters*, all relative pruning parameters have to be scaled as well.

This is exactly the behavior which can be observed in the top plot of Figure 7.6. The figure shows the correlation between different language model and relative pruning parameters with respect to WER. All topN pruning parameters were kept unchanged. As expected, if the relative pruning parameters were kept unchanged as well, the WER increases as lz increases. However, if the relative pruning parameters are increased as well, the WER improves as lz increases. The influence of the word penalty on the overall recognition accuracy is rather small.

On the other hand, the more the relative pruning parameters and lz are increased, the less the considerations of variations in the acoustic model score during pruning. As a result, more acoustic model scores will be computed and decoding is getting slower. The lower plot of Figure 7.6 confirms this. If lz is increased without changing the relative pruning parameters, the real-time-factor decreases. If the relative pruning parameters are increased as well, the real-time factor increases.

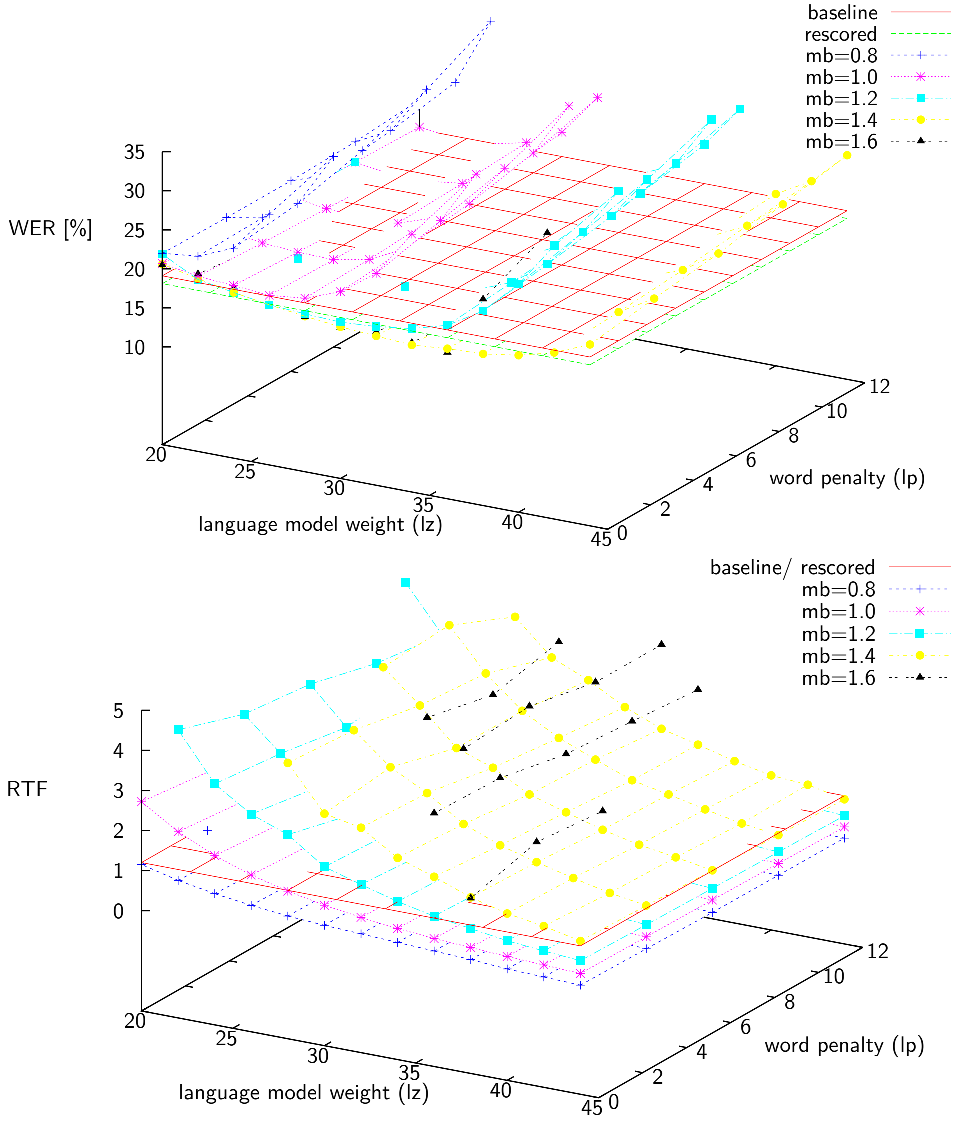


Figure 7.6: The relationship of different pruning and language model parameters and their influence on error rate (WER) and decoding speed (RTF) in Ibis. For the baseline, decoding was done with default pruning and language model parameter settings.

When comparing both plots, it can be seen that although the same WER can be achieved with other values of the relative pruning and language model parameters, all settings with increased parameters result in an increase in RTF. Therefore, a two pass approach seems to be advantageous, in which first, smaller language model parameters are used for the decoding, so that the acoustic model dominates the search process, and second, based on a lattice a rescoring is performed with higher language model parameters in this reduced search space. As can be seen from the results presented in the figures, after language model rescoring the WER is decreased by 1% absolute with an immeasurable increase in RTF only.

7.3 Gaussian Selection in Ibis

From the experimental results in the section above, we have seen that there is a break point in the pruning parameter space from which the relationship between decoding speed and accuracy drastically degrades, i.e. any additional increase in decoding speed results in a disproportional increase in WER. Therefore, to further increase the decoding speed other techniques, which can be used in addition to search-space pruning are necessary.

As already described in Section 4.3.4, in our case the baseline acoustic model consists of about 4000 context dependent codebooks, each modeled with a maximum of 64 Gaussians over a 42-dimensional feature space. For a given input vector $\mathbf{o}$ and a single allophone[2] modeled by an n-dimensional Gaussian mixture model, the scoring function for computing the observation probabilities is defined as follows:

$$f(\mathbf{o} = o_1, \ldots, o_n) = \sum_k \omega_k \frac{1}{(2\pi)^{n/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{o}-\mu_k)^T \Sigma_K^{-1} (\mathbf{o}-\mu_k)}$$

given that the Gaussian mixture model is represented with k components and each component with mean μ_k, a covariance matrix Σ_k, and mixture coefficients ω_k. For the baseline acoustic model, this means that $k \leq 64$ and $n = 42$. Due to precision problems when multiplying a huge amount of very small values, the observation probability computation has to be done in the log-domain and as a first speed-up only diagonal covariances are commonly used, which results in the following equation:

$$\log f(\mathbf{o} = o_1, \ldots, o_n) = \log \sum_k e^{\log \omega_k - \frac{1}{2}\left(\log((2\pi)^n) + \log(|\Sigma_k|) + \sum_{i=1}^n \frac{(o_i - \mu_{ki})^2}{\sigma_{ki}}\right)}$$

Since the sum over all components is computationally very expensive, commonly only the component for which the Mahalanobis distance to the current input vector $\mathbf{o}$ is the smallest is evaluated as an approximation (*Nearest*

[2] Models for phonemes in the context of one or more adjacent phonemes are called polyphones. If the same model is used for several similar polyphones, these models are referred to as allophones.

Neighbor Approximation, [Wos98]). This still requires that all Mahalanobis distances be computed for all k. Since we are interested in only the smallest one, the question arises if it is possible to limit the number of components to evaluate only those which are most likely to achieve the smallest Mahalanobis distance. For selection, the input vectors can be used. While search-space pruning reduces the amount of transitions from all active states of one input vector to all possible states of the next input vector, Gaussian selection reduces the amount of Mahalanobis distance computations necessary per input vector. In the following sections, two different Gaussian selection techniques are explained and compared with respect to their WER vs. speed-up ratio.

Compared to [OFN97, GS06], in which similar studies were presented, the focus of the following experiments is to analyze the influence of different acoustic model sizes on Gaussian selection performance. Furthermore, the current implementation in Ibis for Gaussian clustering was improved, so that the overhead of this technique was reduced and Bucket Box Intersection outperformed.

7.3.1 Bucket Box Intersection

The Bucket Box Intersection algorithm [FR96] defines for each Gaussian in a codebook a rectangular box around the ellipsoid where the value of the Gaussian distribution falls below a certain threshold γ. If the K-dimensional input vector $\mathbf{o}$ does not fall into the Gaussian box of a codebook, this vector can be ignored when computing the observation probability for $\mathbf{o}$. All Gaussian boxes are organized in a K-dimensional space partitioning tree (K-d tree) [Ben75], where each question of the decision tree represents a hyper-plane in a K-dimensional space perpendicular to one axis. Such a hyper-plane can be described by the intercept of the hyper-plane with this axis and divides the space into two half-spaces. This means that a tree with a depth d divides the feature space into 2^d rectangular regions (buckets). This allows for any input vector $\mathbf{o}$ to easily locate the bucket, i.e. leaf of the K-d tree with the Gaussians necessary to evaluate by a sequence of d scalar comparisons.

In the original version of the BBI algorithm, a single tree was built per codebook. In the generalized version [Wos98], the one which is used in this thesis, the decision tree is built over the Gaussians of all codebooks. Each Gaussian is assigned to every bucket with which the Gaussian box intersects. If a bucket does not contain any Gaussians of a certain codebook, the nearest Gaussian of the codebook according to an Euclidean distance is assigned to the bucket. This means that each bucket contains at least one Gaussian of each codebook as back-off.

acoustic model (AM)	Gaussians per frame	percentage of Gaussians
3000-64	35130	19%
3000-128	65812	20%
3000-256	106742	21%
4000-64	36492	16%
4000-128	66177	17%
4000-256	98527	18%
6000-64	37087	12%

Table 7.3: For each acoustic model not using Gaussian selection, the average number of Gaussians evaluated per frame during decoding of *lectDev* and its percentage with respect to the total number of Gaussians of the acoustic model (see Table 7.1).

Experiments

To analyze the performance of the BBI algorithm, trees with different depths d and thresholds γ computed for different acoustic models were compared. The performance was measured using the correlation between word error rate and average number of Gaussians evaluated per frame. The overhead for finding the bucket for a given input vector, which depends on the tree depth, is ignored in this analysis, because we were interested in the quality of the BBI algorithm itself. Later, when comparing different Gaussian selection approaches, their overhead will be considered.

Table 7.3 shows the average number of Gaussians per frame (GPF) which need to be evaluated during decoding for different acoustic models without using BBI. It also shows the proportion of the total number of Gaussians that this represents. The absolute number of Gaussians evaluated per frame increases with the number of Gaussians per codebook and, usually, also with the number of codebooks per acoustic model. The reason why this is not true for the *4000-256* compared to the *3000-256* AM lies in the training. With incremental growing of Gaussians, each Gaussian is split into two based on the number of available training samples per Gaussian, and a threshold is used to guarantee that each Gaussian is trained on a sufficient amount of samples. Furthermore, it can be seen that the percentage of Gaussians evaluated per frame correlate positive with the number of Gaussians per codebook, but negative with the number of codebooks.

From the results presented in Figure 7.7, it can be seen that increasing γ decreases the number of Gaussians per frame (GPF) for a specific tree depth, but increases the WER as well. The same is true for an increase in tree depth, which also results in a decrease in the number of GPF together with an increase in WER. By comparing the number of GPF of the systems not using BBI with those in Figure 7.7, it can be seen that as the reduction gets larger, the number of Gaussians in the acoustic model grows. By using

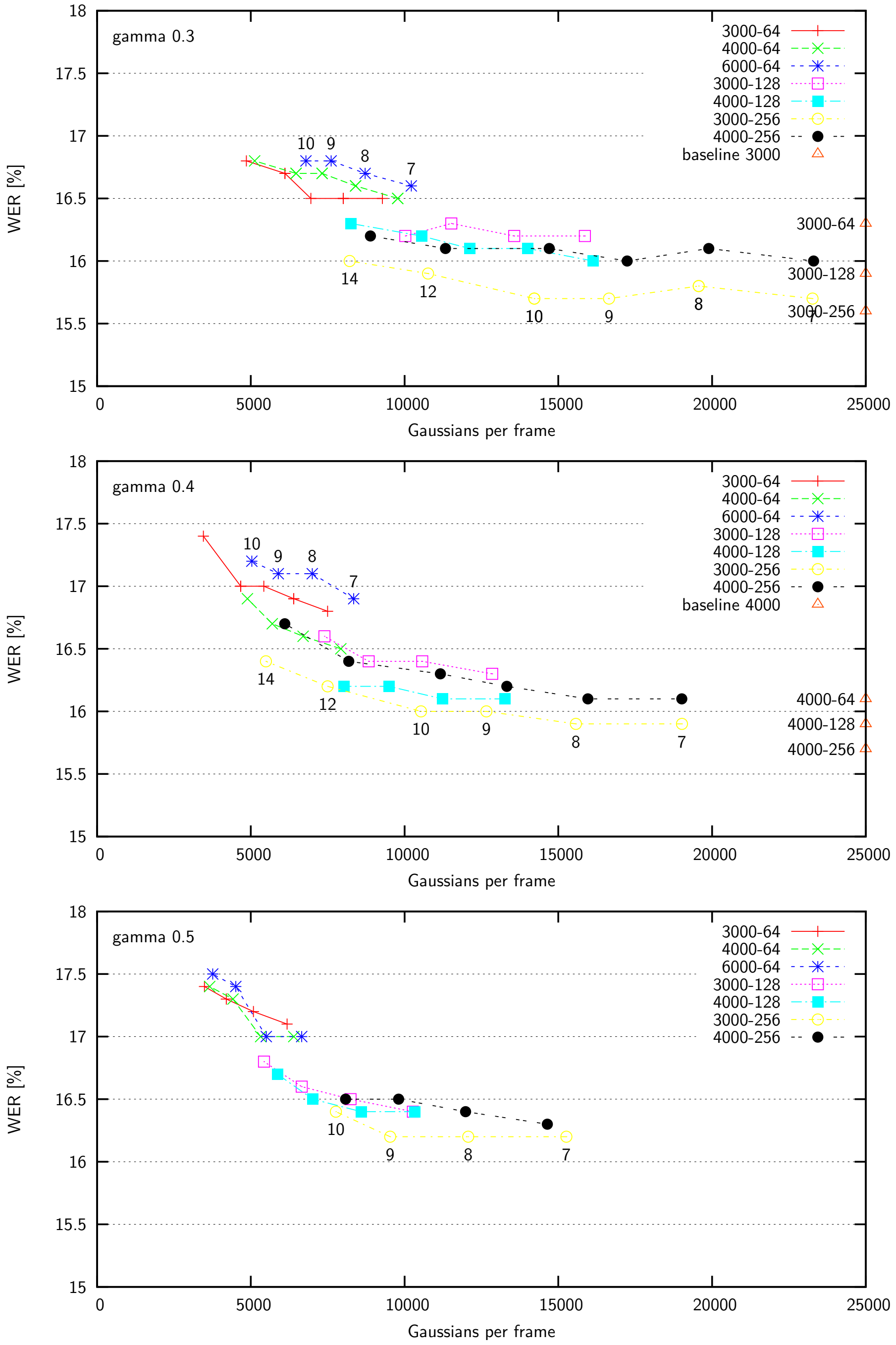


Figure 7.7: Analysis of the correlation between word error rate and average number of Gaussians to evaluate per frame by using the BBI algorithm with different tree depths and thresholds γ. The depth is shown by the number beside the curve points. On the right border line, the baseline WERs without using BBI are given. All results are computed on *lectDev*.

BBI with a tree depth of 10 and a γ of 0.3, the number of GPF is reduced to 17% of the GPF used without BBI for the *3000-64* AM. For the *3000-256* AM, the number of GPF is reduced to 13% only. There is only a minor increase in the WER. For the *3000-64* AM, the WER increases by 0.4% absolute and for the *3000-256* AM by 0.1% only. At almost the same number of GPF, i.e. at a tree depth of 14 for the *3000-256* AM and a depth of 8 for the *3000-64* AM, the *3000-256* AM is still 3% relative better. The higher the γ threshold and/ or the tree depth, the smaller the difference in WER between different AMs.

Overall, it can be seen that the performance for systems with 3000 or 4000 codebooks and the same maximum number of Gaussians per codebook is similar, with the systems with 3000 codebooks tending to perform slightly better. And when comparing the error rates achieved with different γ but identical acoustic models, with a γ of 0.3 the smaller error rates at the same number of GPF can be achieved.

7.3.2 Gaussian Clustering

The BBI algorithm mentioned above has one major disadvantage especially on devices where memory is small. Due to the fact that from each codebook at least one Gaussian has to be present in each bucket, a BBI with a tree depth of 10 and $\gamma = 0.3$ requires about 50MB of memory. Since memory requirement is also a problem for the lecture translation system due to the many different components, the performance of a low memory consumption technique was analyzed and compared with the BBI algorithm.

In Gaussian clustering [Boc93, KGY96, WSTI95, SPZ05], the feature space is partitioned into cells, for each of which a centroid is computed. In computing the observation probabilities, only those Gaussians are evaluated for which the centroid is closest to the input vector. Existing Gaussian clustering methods vary in the techniques used for clustering (k-means, top-down), the distance measure used (Euclidean, Mahalanobis, KL-divergence), or in the way Gaussians are assigned to clusters (disjoint, overlapping). Especially this last characteristic influences the number of clusters activated for a given input vector. During decoding, typically only a fixed number of clusters are activated for a given input vector. Those codebooks for which an observation probability is required but for which no Gaussians are in the activated clusters, a back-off value has to be used.

In our case, we use a k-means-like clustering, in which we start with a random number of centroids usually $\frac{1}{4}$ of the final number of clusters. All clusters are disjoint, i.e. each Gaussian belongs to the closest centroid only. In each iteration, the largest cluster, i.e. the cluster with the largest number of Gaussians, is split into two and new centroids are computed. Furthermore, for all clusters with less then 10% of the average number of Gaussians over all clusters, the centroid is removed and the Gaussians are assigned to the next closest centroid. This is done until the final number of clusters is reached. The Euclidean distance was used, as the distance measure for clustering. For decoding, the Mahalanobis distance was used to select the top N clusters. For backing-off, a fixed value of 256 was used.

Optimizing the probability computation

One advantage of Gaussian clustering is that the amount of memory is much lower than that needed by BBI. 1.5MB is typically sufficient — even for large acoustic models. The disadvantage of Gaussian clustering is the overhead of the distance computation for activating the top N clusters compared to the BBI in which only a few scalar comparisons have to be made. On the other hand, we will see that the quality of Gaussians in the selected clusters is much higher compared to BBI, and therefore a smaller amount of Gaussians have to be activated to achieve the same performance.

To further speed-up the observation probability estimation, the code of the scoring function was optimized with respect to the Gaussian clustering. For normal observation probability computation, as well as when using BBI, the allophones for all possible state transitions are collected and given to the scoring function. The scoring function then loops over the allophones, computes the Mahalanobis distance for all Gaussians in a codebook belonging to the allophone and computes the observation probability for only the closest Gaussian given the input vector. When using BBI, only those Gaussians of a codebook are evaluated which are found in the corresponding leaf node.

This organization is disadvantageous for Gaussian clustering, because, due to the disjointedness of the clusters, active Gaussians cannot be found for all necessary allophones. Furthermore, for all Gaussians of a given allophone, it has to be checked, if the Gaussian is in an active cluster or not, and if not, a back-off value has to be used. For Gaussian clustering, similar to [SZK+03], the observation probability computation was reorganized in the following way.

1. The set of active clusters is determined.

2. The Mahalanobis distances for all necessary allophones are first initialized with a pre-computed back-off value.

3. For all Gaussians in all active clusters, the Mahalanobis distance is computed and compared with the back-off value of the allophone to which the current Gaussian belongs to. If the new distance is smaller, it is remembered for further comparison.

4. The final observation probabilities are computed for all necessary allophones using the smallest Mahalanobis distances.

As can be seen in Table 7.4, the real-time factor is reduced by 6.25%, due to the optimized probability computation.

	WER	RTF
using Gaussian Clustering (GC)	17.1	0.80
using GC with speed-up	17.1	0.75

Table 7.4: Comparison between the standard and the optimized observation probability computation on *lectDev*.

Experiments

As for the experiments for the BBI algorithm, different parameters used for Gaussian clustering were compared on different acoustic models with respect to their performance, i.e. WER and GPF. The overhead due to the activation of the clusters is ignored in this first comparison, because we are interested in the quality of the clustering itself. Later, when comparing different Gaussian selection approaches, the overhead of those will be included. The total number of clusters to which all Gaussians have to be assigned and the number of active clusters used during decoding are the two parameters which we explore.

Since the number of Gaussians per cluster decreases with an increasing total number of clusters used for clustering (*clusterN*), it can be seen from Figure 7.8 that the number of GPF decrease as well. But at approximately the same level of WER, the number of Gaussians evaluated per frame is smaller for the systems using 1024 total clusters instead of 2048 clusters, when comparing the same acoustic models. Furthermore, the gradient increases more with a decrease in the number of active clusters (*topN*) for 2048 compared to 1024 total clusters.

Compared to the total number of GPF needed to evaluate for the baseline systems without Gaussian clustering (see Table 7.3), it can be seen that for the same number of active clusters the number of GPF can be reduced to almost the same percentage of GPF used without any Gaussian selection for all tested acoustic models. For example, for 64 active clusters (out of 1024), the number of GPF can be reduced to 11-12% for all tested AMs. The increase in WER in that case is only about 1-2%. Furthermore, it can be seen that there is almost no difference in performance for systems with 3000 or 4000 codebooks. When comparing WERs at the same number of GPF, it can be seen that up to about 7000 GPF the AMs with a smaller number of Gaussians perform better than those with a higher number of Gaussians. But after this break-point, the relationship is inverted.

7.3.3 Comparison and Discussion

Figure 7.9 compares the BBI algorithm with Gaussian clustering with respect to their speed-up and real-time performance, i.e. by including the overhead for identifying the Gaussians to be evaluated for a given frame. As

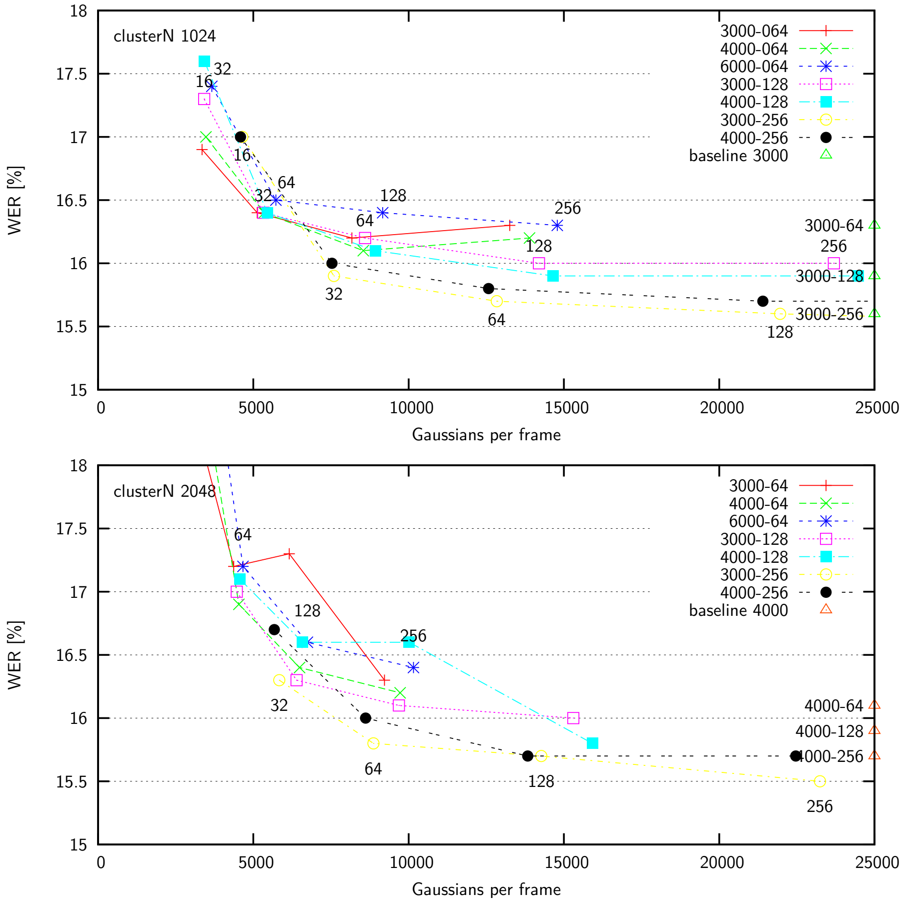


Figure 7.8: Analysis of the correlation between word error rate and average number of Gaussians to evaluate per frame by using Gaussian clustering with different numbers of total and active clusters. The number of active clusters is shown beside the curve points. On the right border line, the baseline WERs without using Gaussian clustering are given. All results are computed on *lectDev*.

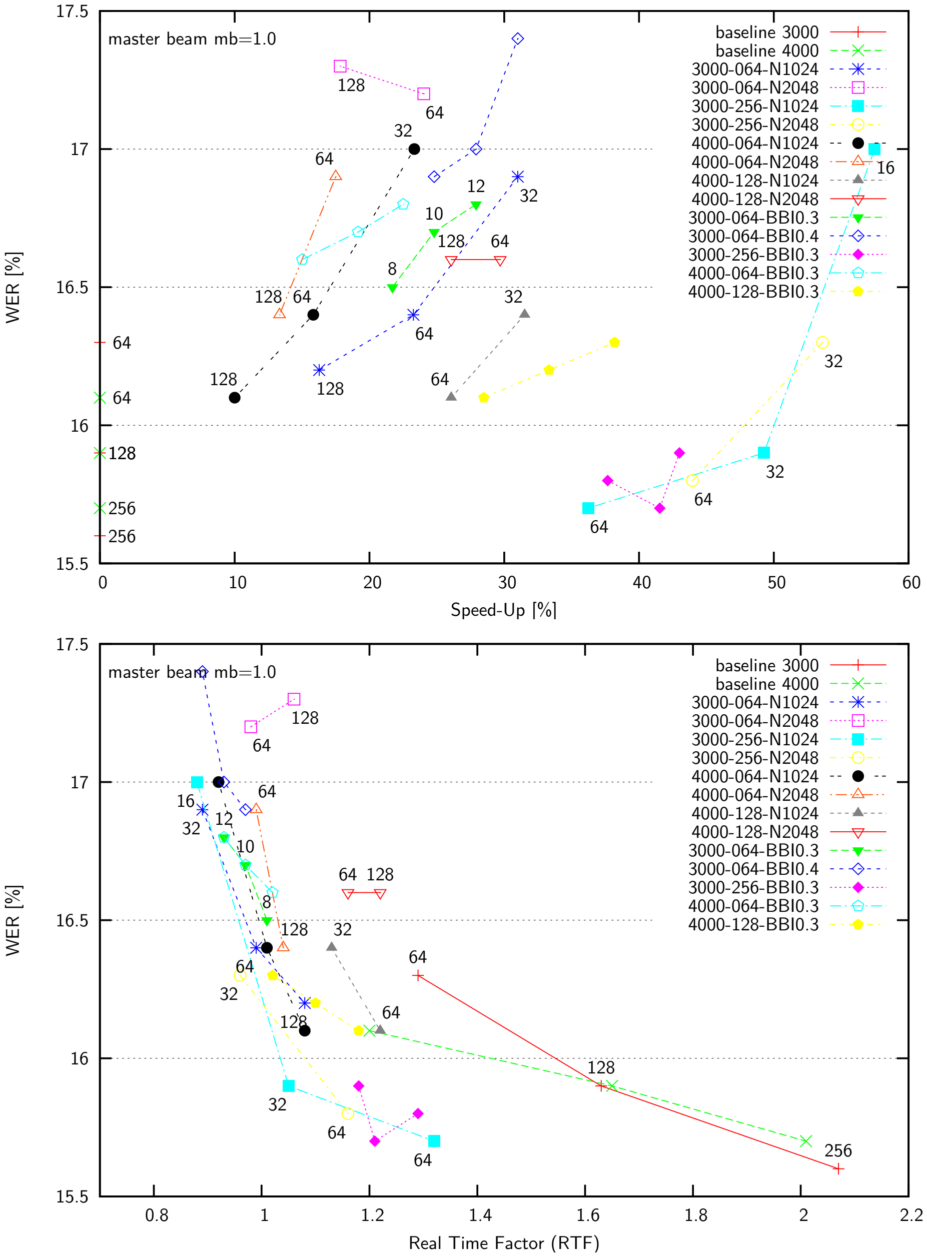


Figure 7.9: Comparison of the performance of different Gaussian selection techniques. In the top figure, the speed-up relative to the baseline system is given; in the bottom figure the real-time factors are compared. For the baselines, the numbers shown beside the curve points represent the maximum number of Gaussians per codebook. For the other curves, they represent either the depth of the BBI tree (8, 10, 12) or the number of activated clusters (16, 32, 64, 128). Some of these numbers are omitted because of clarity reasons.

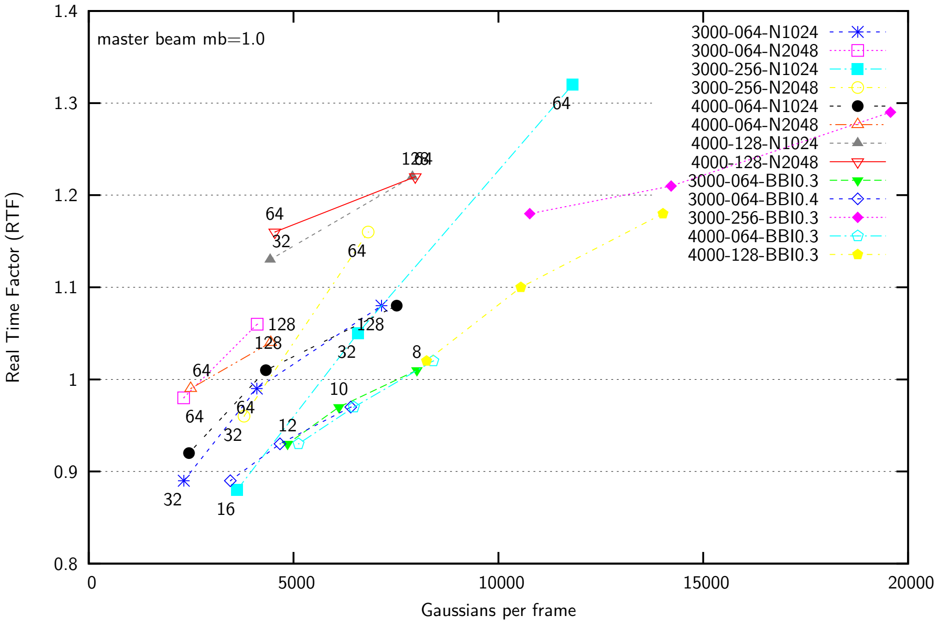


Figure 7.10: Comparison of the real-time factor as a function of the numbers of Gaussians evaluated per frame for different Gaussian selection techniques.

can be seen for both techniques, there is a substantial speed-up compared to the baseline which does not use any kind of Gaussian selection. At almost the same word error rate, a system using Gaussian clustering is about 10%-36% (*4000-64, 3000-256*) faster; the speed-up is larger the higher the total number of Gaussians in the AM. For systems using the BBI algorithm, the maximum speed-up is even as high as 41% (*3000-256*). However, when comparing the performance of the BBI algorithm and Gaussian clustering in general, it can be observed that for most systems slightly better recognition results can be obtained at the same decoding speed with Gaussian clustering. Another interesting observation is that the best performance regarding the relationship between decoding speed and recognition accuracy can be achieved with the *3000-256* AM using Gaussian clustering with a total of 1024 clusters. The associated curve is the one closest to the bottom right corner of the bottom plot of Figure 7.9. At a real-time factor of almost 1.0 (1.05), the WER compared to the baseline increases by only 1.9% (from 15.6% to 15.9%). But the distance in terms of WER and RTF to other acoustic models is getting smaller, the smaller the obtained real-time factor.

Figure 7.10 compares the real-time factor as a function of the numbers of Gaussians evaluated per frame for different AMs using different Gaussian selection techniques. First of all, it can be seen that for all AMs and Gaussian selection techniques, this relationship is almost linear, but at the

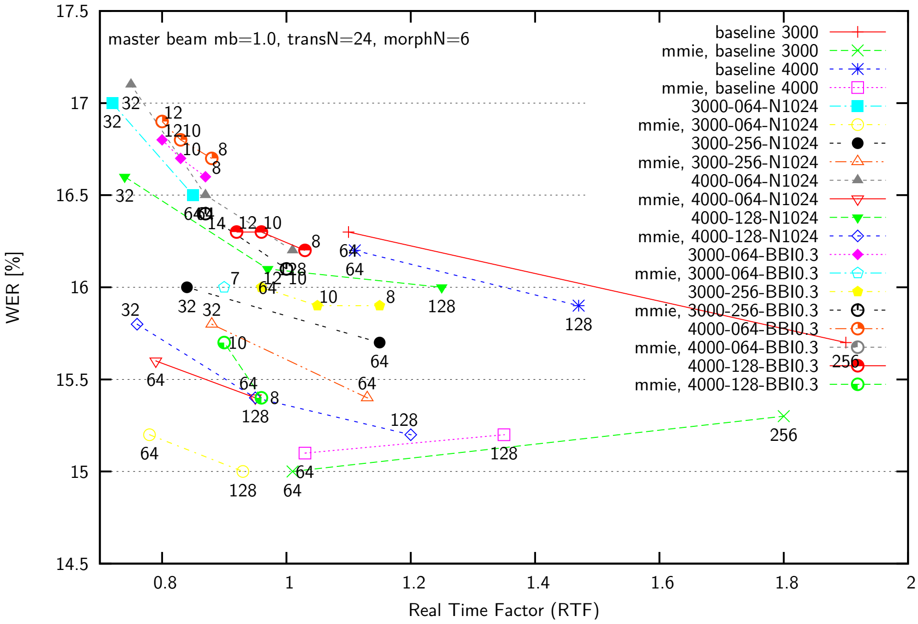


Figure 7.11: Comparison of the results for non-discriminatively and discriminatively (MMIE) trained AMs using either Gaussian clustering with 1024 clusters or the BBI algorithm with a γ of 0.3.

same number of GPF some techniques have a lower RTF than others. There exist two explanations for this behavior. First, the techniques may differ in the quality of Gaussians selected and therefore during decoding a smaller search space is explored. Second, some Gaussian selection techniques have smaller overhead than others. While the results using the BBI algorithm fall along a straight line, the results using Gaussian clustering depend more on the number of clusters or Gaussians in the AM. Furthermore, it can be seen that to achieve the same real-time factor, a smaller amount of Gaussians must be evaluated per frame for Gaussian clustering than for the BBI algorithm. But when comparing this figure with Figure 7.9, it can be seen that although the amount of Gaussians evaluated per frame for the AMs using Gaussian clustering with 1024 total clusters is smaller, the WER is often lower than compared to the BBI algorithm with a tree depth of 0.3 for the same real-time factor. This clearly shows the higher overhead and the higher quality of Gaussians selected by Gaussian clustering.

So far, all acoustic models used for the experiments above were trained up to step 5 (see Section 4.3.4). In the following experiments, it was analyzed if and how the obtained results from above change when discriminative training (MMIE) is applied. Three observations can be made from Figure 7.11. First of all, better results are now obtained with acoustic models using a smaller number of Gaussians. The smallest WER is achieved with *3000-*

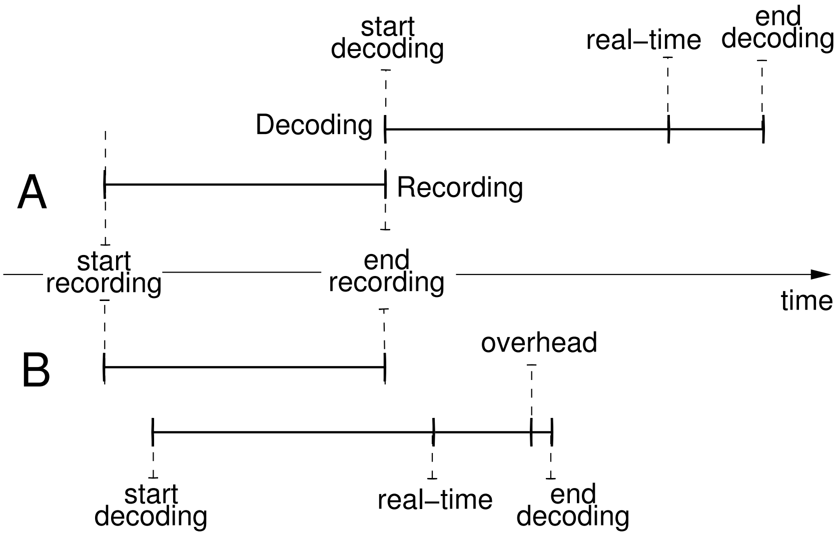


Figure 7.12: Schematic overview of the latencies produced by an offline decoding strategy (A) and a run-on decoding strategy (B) with a system running slower than real-time.

64 AM. Second, as a result of the improvements due to the MMIE training, the real-time factor also decreases. And third, the improvements in performance by using Gaussian selection for the discriminatively trained AMs are smaller than compared to those obtained with non-discriminatively trained AMs, and are smaller for the BBI algorithm than for Gaussian clustering.

7.4 Limiting the Latency

As already stated in the introduction to this chapter, overlapping processing is necessary to keep the latency as small as possible. For this purpose, the most common strategy for speech recognition is the so-called *run-on decoding strategy*, in which decoding starts before the current utterance is complete. Figure 7.12 clarifies the difference between an offline and a run-on decoding strategy. As can be seen in the case of an offline decoding strategy (A) decoding starts directly after the recording has finished. The latency is the sum of the recording and the decoding time and therefore depends on the decoding speed only. For a run-on decoding strategy (B), the overall latency depends on the amount of time which the recognizer has to wait until a sufficient amount of data is available, as well as on the decoding speed, and on additional overhead due to the more complicated processing. Nonetheless, the latency can be significantly reduced by this strategy. The most prominent example of run-on decoding are commercial dictation systems.

In a basic run-on decoding strategy, the final hypothesis is returned after the recording as well as the decoding have been finished. If the recording is short, this is not a problem. For lectures, in which a more or less continuous stream of speech is recorded, this might be more problematic. Different solutions for this problem exist:

- A voice activity detector (VAD) is used to partition the recording in chunks small enough to guarantee short latency.
- In addition to a less sensitive VAD, the decoder is modified to return partial hypotheses in short intervals while decoding the current recording.

In the first case, the VAD tries to detect non-speech regions to segment a recording, so that the recognizer is able to finish the current decoding and to return the final hypothesis. Therefore, the decoder does not have to be modified, and decoding time can be saved because non-speech regions are already skipped by the VAD. However, finding appropriate split points might be difficult, especially when a speaker speaks fluently and quickly, and small recording chunks which are necessary to keep the latency low can not be guaranteed. On the other hand, if break points do correlate well with sentence or semantic breaks, so that the subsequent resegmentation becomes superfluous, using a VAD in front of speech recognition would still make sense. As we will see later in Chapter 8, the segments produced by a VAD using acoustic features only are worse in terms of translation performance than when linguistic features are used in addition.

For this reason, the second solutions in which the speech recognizer returns partial hypotheses already during decoding of the current recording was preferred for the lecture translation system. In this case, only a rough segmentation is provided by a simple energy based VAD to prevent overflows in the decoder. Furthermore, a resegmentation component for merging and resegmenting the partial hypotheses delivered from the speech recognizer, so that optimal machine translation performance together with low latency are achieved, is now essential. In the following, the necessary changes will be described in more detail.

7.4.1 Partial Trace-Back

At a given point in time during decoding, typically more than one active paths exist in the search-space and will be expanded when processing the next frame. This poses the question of how to return a partial hypothesis. For example, one can select the best hypothesis based on the likelihood, but, due to search-space pruning, even the most likely hypothesis can be pruned away in the future, and another hypothesis may become the first best. But this new first-best hypothesis can be different from the old one. In order to prevent mis-translations in the simultaneous translation system, one has to make sure that only those partial hypotheses are returned by the speech recognizer which will not change in the future.

To support this, a *partial trace-back* mechanism was implemented in Ibis. At a given point in time during decoding, all active paths in the search-space are traced back, and as soon as the trace-backs become unique the words along the unique path can be returned as a first-best partial hypothesis. Subsequent partial trace-backs always return the words along the unique path starting from the end point of the preliminary trace-back, up to the point where the search path diverges. Since the number of active paths in each frame is limited – by the *transN* pruning parameter – the additional overhead is relatively small. Unfortunately, further optimization of the speech recognizer's output through lattice rescoring is no longer possible. Due to edges added to the lattice after the decoding has been finished, lattice rescoring is non-linear with time.

7.4.2 Front-End and Online Adaptation

As described in Section 4.3.3, audio offset correction, cepstral mean subtraction (CMS), and cepstral variance normalization (CVN) are performed via global normalization over the whole utterance. But in combination with the partial trace-back, utterance based normalization is no longer possible. Instead, normalization parameters have to be estimated from the partial hypotheses and have to be updated incrementally. Furthermore, an additional history weighting was introduced so that the normalization can adapt to changing environmental acoustic conditions.

In order to judge the amount of data necessary for updating the normalization parameters and the influence on latency of the additional overhead of the update as well as the partial trace-back, a few experiments with different interval lengths were performed. As can be seen in Table 7.5, the smaller the interval, i.e. the more often the parameters are updated based on the partial trace-back result, the higher the overhead. While for an interval with five seconds a total real-time factor of 0.93 was measured, for an interval of 0.5 seconds it was 1.05. The total real-time factor includes, in addition to the decoding, also the time necessary for updating the normalization parameters and for the partial trace-back. Furthermore, the shorter the interval, the shorter the latency. In terms of WER, it can be seen that there is an optimum at an interval of three seconds. Unfortunately, because of the latency, only intervals between one or two seconds in duration are possible. Interesting in the results is that due to the more frequent update of the normalization parameters, leading to a higher sensitivity against changes in the environmental acoustic conditions, the baseline WER can be outperformed.

The baseline is a system which uses manual segmentation and utterance-based global normalization in the front-end. No other kind of adaptation

interval length	WER	LAT	RTF decoding	RTF total
baseline	16.9%		1.10	1.10
0.5	17.3%	3.48	1.01	1.05
1.0	16.7%	3.60	0.97	0.98
2.0	16.6%	5.10	0.95	0.96
3.0	16.4%	6.83	0.94	0.95
5.0	16.6%	10.36	0.93	0.93

Table 7.5: Comparison of the word error rate, latency in seconds, real-time factor of the decoding itself and the total real-time factor for different audio interval durations (in seconds) on *LectDev*.

was performed during decoding and no lattice rescoring was applied, for all experiments. The baseline system is identical to the one described in Section 5.3, which achieved a WER of 16.4% after lattice rescoring (see Table 5.1, column VTLN-AM).

The same changes are necessary for online adaptation. For both VTLN and cMLLR, an additional history-weighting factor has to be introduced. Table 7.6 compares the results of the baseline system using manual segmentation with systems using partial trace-backs after intervals of 1-2 seconds. As can be seen, the systems' recognition accuracy suffers only slightly. VTLN improves the WER only slightly but does not influence latency or the real-time factor. In contrast, cMLLR reduces the WER significantly, but the total RTF increases as well. On the other hand, the decoding is accelerated due to adaptation, which can be observed in the smaller decoding RTF when cMLLR is applied.

Latencies achieved for the systems using a one second interval are significantly lower than those obtained with a two second interval if the total real-time factor is lower than one. In the case of the system with incremental VTLN and cMLLR using a one second interval, it can be seen what happens if the total real-time factor is higher than 1.0. The system cannot keep up with the speaking rate of the input speech leading to high latency. In such cases, applying additional speed-up techniques as described above are essential. The results can be seen in the conclusion to this chapter.

7.4.3 Language Model

For language model training, the training corpora are traditionally split at sentence boundaries in order to be able to estimate transition probabilities for words at the beginning or end of a sentence. But due to continuous processing of the input speech, together with a trace-back, this reformatting no longer makes sense. Therefore, in accordance with [SCER97, GLA98]

	WER	LAT	RTF	RTF
			decoding	total
baseline	16.9%		1.10	1.10
+ VTLN	16.6%			
+ VTLN + cMLLR	15.1%			
1.0	16.7%	3.60	0.97	0.98
+ VTLN	16.4%	3.60	0.95	0.98
+ VTLN + cMLLR	15.3%	6.33	0.86	1.18
2.0	16.6%	5.10	0.95	0.96
+ VTLN	16.3%	5.10	0.93	0.95
+ VTLN + cMLLR	15.3%	5.80	0.85	1.08

Table 7.6: Comparison of the baseline system using manual segmentation with systems using partial trace-back at intervals of 1-2 seconds of input speech. Results after applying VTLN as well as cMLLR are compared in terms of word error rate (WER), latency (LAT) and real-time factor (RTF) on *lectDev*.

	sentence LM	continuous LM
baseline	16.9%	16.9%
2.0	16.7%	16.3%

Table 7.7: Comparison of the results (WER) obtained by using a sentence language model and a continuous language model with the baseline system, i.e. with manual segmentation, and a system using the partial trace-back mechanism with an interval of two seconds. All experiments were performed on *lectDev*.

additional n-grams across sentence boundaries were added to the language model. Table 7.7 shows the improvement in WER for the continuous language model if the partial trace-back mechanism is used.

7.5 Conclusion

In this chapter it was discussed how the real-time and latency requirements of a simultaneous lecture translation system can be fulfilled. The focus was to speed-up the computationally most expensive component, the speech recognition, and to introduce overlapping processing into the framework to reduce the latency.

First of all, the search space pruning technologies (relative and topN) implemented for different levels (state, linguistic, and word level) in the Ibis speech recognition decoder were analyzed in more detail. It was investigated how the different pruning parameters influence the speed and accuracy of

	WER lectDev	LAT	RTF decoding	RTF total
baseline 4000-64	15.1%		1.14	1.20
3000-64	14.9%		1.14	1.20
+ optimized pruning	15.2%		0.88	0.93
+ Gaussian clustering	15.4%		0.65	0.70
1.0	15.5%	2.16	0.56	0.88
2.0	15.2%	3.61	0.55	0.78

Table 7.8: Final performance results after speeding-up the decoding and reducing the latency of the speech recognition on *lectDev*. While for the top baseline the *4000-64* AM was used, the other results were obtained with the *3000-64* AM. For the experiment with a trace-back at intervals of one or two seconds, all optimizations from above were used. The results are given in word error rate (WER), latency (LAT) and real-time factor (RTF).

the decoder and how they depend on each other. Changes in the *state beam* influences the ratio between recognition accuracy and decoding speed most, followed by the *word beam*. The influence of the topN pruning parameters is only little. As a result of this analysis, an optimized pruning parameter combination was determined.

It was shown that although Ibis is a single pass decoder, further language model rescoring based on lattices produced during decoding improves the WER with only a minor increase in overall decoding time. In this context, the relationship between language model parameters and pruning parameters was explained. For the *3000-64* AM, the real-time factor could be improved by 24% relatively to 0.88 with an increase in WER by 1.4% relative to 14.6% only.

In the next step, optimizations in the Gaussian Layer of the Ibis decoder were investigated. Two algorithms, the Bucket Box Intersection (BBI) algorithm and Gaussian clustering, were compared with respect to their performance, i.e. decoding speed vs. accuracy. For both methods, the results were compared for different acoustic models and with different parameter settings. It turns out that, with appropriate tuning, Gaussian clustering outperforms BBI in terms of speed and accuracy. Therefore, the function for computing the observation probabilities was optimized to reduce the overhead of the Gaussian clustering. Another observation was made after discriminative training was applied to the acoustic models. The best results are now achieved with AMs using a smaller number of Gaussians. Also, the improvement in performance by using Gaussian selection for the discriminatively trained AMs is smaller than compared to that obtained with non-discriminatively trained AMs.

Tables 7.8 and 7.9 summarize the results. First, the baseline system us-

	WER lectEval	LAT	RTF decoding	RTF total
baseline 4000-64	16.6%		1.21	1.28
3000-64	16.2%		1.23	1.30
+ optimized pruning	16.3%		0.94	1.00
+ Gaussian clustering	16.8%		0.70	0.77
1.0	17.0%	2.95	0.61	0.93
2.0	16.9%	4.30	0.60	0.83

Table 7.9: Final performance results after speeding-up the decoding and reducing the latency of the speech recognition on *lectEval*. While for the top baseline the *4000-64* AM was used, the other results were obtained with the *3000-64* AM. For the experiment with an trace-back at intervals of one or two seconds length all optimizations from above were used. The results are given in word error rate (WER), latency (LAT) and real-time factor (RTF).

ing the *4000-64* acoustic model is compared to the *3000-64* acoustic model. Both are trained with MMIE and use default pruning parameters, no Gaussian selection, and incremental adaptation during decoding. Second, the optimized pruning parameters were applied to the *3000-64* AM and then Gaussian clustering was used in addition. For Gaussian clustering, the best 64 clusters out of 1024 were selected in each frame.

As can be seen, the *3000-64* AM performs slightly better than the *4000-64* AM at the same speed. When using the optimized pruning parameters as well as the Gaussian clustering, the real-time factor is almost halved, with an increase in WER of 3.4% on the development data and only about 4.3% on the evaluation data. Furthermore, by using the partial trace-back mechanism together with the continuous language model, the latency could be reduced to 2-3 seconds for a one second interval and to about four seconds for a two second interval. The latencies on the evaluation data are higher than those on the development data, which is related to the higher WER and RTF. Due to the much lower latency, the one second interval is preferred over the two second one even if the WER is slightly worse.

Overall, a WER of 15.5% with a latency of 2.16 seconds could be achieved on the development data and a WER of 17.0% with a latency of 2.95 seconds on the evaluation data. The obtained latencies hold for speech recognition only. In the next chapter, the interface between speech recognition and machine translation will be explored.

Chapter 8

Translatable Speech Segments

In speech translation systems, the combination of automatic speech recognition (ASR) and machine translation (MT) is not always straight forward when optimal performance is the goal. For the lecture translation system, in addition to the errors committed by the speech recognition leading to errors in machine translation, the partial ASR hypotheses have to be resegmented such that MT performance does not suffer further. Since almost all MT systems are trained on data split at sentence boundaries, this is commonly done by resegmenting the hypotheses according to automatically detected sentence boundaries.

However, automatic sentence boundary detection, or punctuation annotation in general, is, depending on the type of data, still very challenging. Punctuation annotation is usually done by combining lexical and prosodic features [Liu04], and the combination is often done with the help of maximum entropy models [HZ02] or CART-style decision trees [KW01].

Within TC-STAR, Lee et al. [LAOPR06] proposed a system which inserts commas within a given ASR sentence by using n-gram statistics for commas together with certain thresholds to improve MT quality. [RSM+06b] proposed another solution for inserting commas and periods into the ASR output, by using a maximum entropy classifier using durational and language model features. Using this classifiers for each contiguous non-word sequence, it was decided if it is replaced by a comma or period. They observed on English a 98% correlation for periods and a 70% correlation for commas.

In [MMN06], different approaches for automatic sentence segmentation and punctuation prediction were compared with respect to MT performance. Punctuation prediction was either done with the help of a hidden n-gram [Sto98] or by generating them implicitly during the translation process. For sentence segmentation, an HMM-style search using hidden-events to represent segment boundaries was used, extended with an additional sentence length model. To obtain an optimal segmentation of a document, a global search, restricted by the sentence length model has to be performed.

Performing a global search is also necessary for the technique described in [XZN05], where an IBM word alignment model 1 is used, and other approaches to improve example base MT, where longer sentences do not yield good translation performance [KZK00, DS05]. The approach in [LGCL96] splits sentences before and during parsing to improve the translation performance of an Interlingua-based MT system for Spanish-English. The above approaches, however, have focused on limited domain tasks only and are not easily extensible to more difficult domains.

For simultaneous translation systems [FKPW06], merging and chunking of partial ASR hypotheses into useful translatable segments is even more critical and difficult. Due to the resulting latency, a global optimization over several ASR hypotheses, as suggested in the approaches described above, is impossible. Instead, the latency should be kept as small as possible as described in Chapter 3.

This chapter presents and extends the work, inspired by experiments done by Cettolo et. al. [CF06] on investigating the impact on translation performance of different text segmentation criteria, and was published in [FK07]. It addresses the questions of how chunking of ASR hypotheses as well as ASR reference transcripts into translatable segments, usually smaller than sentences, influences MT performance. Different segmentation strategies on ASR hypotheses as well as on the reference transcripts are compared. To measure the usefulness for simultaneous translation, MT performance was set in relation to the average segment length and its standard deviation.

Simultaneously with [FK07], another work was published by Rao [RLS07b]. This work was aimed to improve the translation performance by optimizing the translation segment length only, instead of reducing the segment length to decrease translation system latency. The interesting result of this work is that the best performance is achieved with segments smaller than sentences, similar to the observations described here.

In contrast to Chapter 4, other systems and test data were used, because the translation direction was from Spanish into English instead of vice versa. The reason for this was that to the time when these studies were made, the Spanish-English TC-STAR translation system achieved the best translation results. Therefore, Section 8.1 explains the differences in more detail. The experimental results of this study and a new segmentation algorithm are presented and discussed in Section 8.2. Section 8.3 concludes this chapter and shows that the obtained results can be transferred to translation of lectures in the other direction, i.e. from English into Spanish.

8.1 Data and Systems

As test data, the 2006 Spanish-English TC-STAR development data, consisting of 3hrs (14 sessions) of non-native Spanish speech recorded at the European Parliament, was selected. For this set, ASR hypotheses as well as reference transcripts and translations were available, while the Spanish hypotheses were generated with a system trained within TC-STAR on Parliament Plenary Sessions [SPK+07]. The case-insensitive word error rate was 8.4%.

8.1.1 Statistical Machine Translation

The Spanish-English machine translation system [KZV+06] was trained using minimum error rate training [Och03] on the same parallel EPPS data as mentioned in Section 4.4, but with the translation direction from Spanish to English. For decoding, a word reordering window size of four was used. As will be shown later, the translation quality of this system on the above mentioned data is much higher than the English-to-Spanish results on lectures. The reason for this is the much better match between the training and test data in speaking style as well as topic.

8.2 Experimental Results and Discussion

In this section, the translation scores achieved by translating ASR reference transcripts as well as ASR hypotheses resegmented with different chunking strategies are compared and discussed. Since punctuation annotation of ASR hypotheses is a research problem in itself, and not the focus of this thesis, all punctuation marks in the reference transcripts were removed for comparison reasons. However, the MT system was trained on complete sentences containing punctuation marks, since punctuation marks can provide useful alignment boundaries during the word alignment training.

Another problem is the influence of the LM on the translation quality of different chunking strategies. For language model training, the training corpora are typically split at sentence boundaries so as to estimate the transitions at the beginning and end of a sentence; chunking strategies, which produce segment boundaries with a high correlation to sentence boundaries, are therefore affected adversely. To mitigate this effect, the language model training corpora were resegmented accordingly.

For further analysis, in addition to the translation scores and segment length statistics, precision and recall are computed by aligning the segment boundaries to punctuation marks in the ASR reference transcripts.

8.2.1 Scoring MT with Different Segmentations

The commonly used metrics for the automatic evaluation of machine translation output, such as the Bleu [PRWZ02] and NIST [NIS04] metrics, have originally been developed for written text, where the input segment boundaries correspond to the reference sentence boundaries. This is not the case for translation of spoken language where the correct segmentation into sentence-like units is unknown and must be produced automatically by the system.

In order to be able to use the established evaluation measures, the translation output of the automatically produced segments must be mapped to the reference translation segments in advance, before the scoring procedure. This is done by using the method described in [MLBN05], which takes advantage of the edit distance algorithm to produce an optimal resegmentation of the hypotheses for scoring, and which is invariant to the segmentation used by the translation component.

The Bleu scores presented in this paper were obtained by using this method using two reference translations. Since the alignment was performed on a per-session level, the result is invariant in relation to the number of segments produced for a single session. The scoring was case-insensitive, without taking punctuation marks into account.

8.2.2 Baselines

Resegmenting ASR hypotheses at sentences boundaries for MT is the most common approach for speech translation systems. For this reason, the translation scores obtained by translating ASR hypotheses, as well as reference transcripts split at sentence boundaries (*sent*), serve as one baseline for the following experiments. As can be seen in Table 8.1, a Bleu score of 37.6% by translating ASR reference transcripts, and a score of 34.3% for ASR hypotheses were obtained, which clearly shows the influence of ASR performance on MT quality. The average segment length was 30 words with a standard deviation of 22.

Another baseline is obtained by taking all punctuation marks as split points (*punct*). Here, the average segment length can be reduced to nine words with almost no decrease in the translation score. The reason for this is that punctuation marks are usually used to represent semantic boundaries. However, one needs to be careful because, as the use of punctuation marks differs from language to language, they might be impractical as split points for languages other than Spanish. Note that, for both baselines, the LM training corpus was split accordingly.

In the following sections, it will be analyzed how MT performance is affected by chunking strategies using other features and approaches requiring a smaller amount of context information for their decision.

8.2.3 Destroying the Semantic Context

In the following experiments, it was analyzed how MT performance is affected by destroying the semantic context of an utterance independently of the applicability for simultaneous translation. Following the experiments in [CF06] the merged utterances of a single session were simply cut every n words (*fixed*). The results are given in Table 8.1 for $n \in \{7, 11, 15\}$ and provide a lower bound for the subsequently explored chunking strategies. As expected, the decrease in segment size affected the translation scores dramatically. The translation results could be significantly improved by just cutting a sentence into two (*sent-0.5*) or four (*sent-0.25*) segments of equal size and not splitting across sentence boundaries. This clearly shows the dependency of the MT performance on the segmentation used for training the MT system.

8.2.4 Using Acoustic Features

Following the studies in [Che99] in which pauses were shown to closely correspond to punctuation, the information from non-speech regions in the ASR hypotheses was used for resegmentation. As non-speech regions, recognized silences and non-human noises were used, and successive noises and silences were merged together. For the translation scores (*pause*) in Table 8.1 different non-speech duration thresholds (0.1, 0.2, and 0.3 seconds) were selected. As expected, the results are significantly better than those obtained with the chunking strategies in Section 8.2.3. The precision and recall values clearly validate the studies in [Che99], also for Spanish. While a threshold of 0.1 has the best correlation to punctuation marks, the MT score is the worst.

The standard deviations of the segment lengths achieved with the non-speech chunking strategies are still too large for use in simultaneous translation. By splitting the ASR hypotheses at the longest non-speech interval within a region of a maximum number of words (*var15*, *var20*, *var25*, with chunks of maximal 15, 20, 25 words), the standard deviation could be significantly reduced without decreasing the translation quality when comparing fixed and variable non-speech chunking strategies having a similar average segment length.

Overall, chunking strategies using non-speech regions are simple and require no additional context information, but nonetheless achieve relatively good translation scores. The reason for this seems to be the relatively good correlation of the split points with sentence boundaries, as can be seen in the high precision and recall values.

8.2.5 A New Segmentation Algorithm

Given the results above, a new segmentation algorithm, suitable for the lecture translation system, was developed. Only lexical information, such as punctuation marks, filled pauses and human noises, was taken into account. At least for the languages investigated so far, using additional prosodic features has shown to be less profitable [CF98].

According to [CF98], a trigram language model was trained, in which all lexical cues in the training corpus were substituted by a boundary tag BD. Instead of doing a global optimization over the whole sentence, the decision was made using local information only, by (1) setting the LM probabilities $Pr(w_{i-1}w_iBDw_{i+1}w_{i+2})$ and $Pr(w_{i-1}w_iw_{i+1}w_{i+2})$ in relation to each other; and (2) using additional thresholds for the non-speech gap in between w_i and w_{i+1}, i.e. regions in which the ASR decoder recognized a non-speech event such as silence, a filled pause, or a human noise. Since the LM training data does not contain any acoustic information, it was resegmented by using the lexical cues only.

As can be seen in Table 8.1 (*lm*) this chunking strategy outperforms all other strategies using only acoustic features. A precision of 73% and a recall of 54% were measured. However, a standard deviation of 10 from the average of 11 words was also found. Therefore, in a second experiment the above mentioned thresholds were restricted when no split point was found after ten words (*lm-10*). Thereby, the standard deviation could be almost halved with only a minor decrease in Bleu score. Furthermore, both strategies are slightly better than the baseline computed on sentence boundaries and almost as good as the baseline computed on punctuation marks.

8.3 Conclusion

In this chapter, the question on how utterance chunking influences machine translation performance was addressed in an empirical study by comparing different chunking strategies on ASR hypotheses as well as on ASR reference transcripts. As can be seen in Figure 8.1, sentence boundaries are a good criterion for utterance chunking, but are inapplicable for simultaneous translation because of the high average sentence length. Chunking strategies based on non-speech regions are simple and require no additional context information, achieving relatively good translation scores.

Given these results, a more sophisticated approach using lexical features provided by a LM, in addition to acoustic features, was developed, which outperforms all chunking strategies using only acoustic features and yields similar in translation quality as the baselines. Furthermore, by restricting the thresholds for this heuristic after 10 words (*lm-10*), a suitable average segment length for the lecture translation system of 9 with a small standard deviation of 6 could be achieved.

Table 8.2 shows the results of the experiments, but for the other translation direction, i.e from English into Spanish. Segmentations based on sentence boundaries and punctuation marks in general are compared to the

Chunking	SegLength		Correlation		Bleu	
strategy	avg	sdev	PRC	RCL	Ref	ASR
Baseline						
sent	30.1	22.1	98.5	26.5	37.56	34.28
punct	9.2	6.7	100.0	98.6	38.27	35.47
Destroying the semantic context						
fixed-7	7.0	0.2	22.8	26.5	30.13	27.50
fixed-11	11.0	0.7	22.6	16.8	32.07	29.53
fixed-15	15.0	0.6	23.8	13.0	33.64	30.66
sent-0.5	15.3	11.3	59.0	31.8	35.08	
sent-0.25	10.3	8.5	44.9	36.1	33.67	
Using acoustic features						
pause-0.1	8.2	5.7	59.3	58.3		31.86
pause-0.2	12.1	11.0	66.3	44.5		32.53
pause-0.3	17.0	19.6	71.5	34.0		32.62
pause-var15	7.5	3.1	59.4	61.4		31.34
pause-var20	9.8	4.3	64.6	53.0		31.87
pause-var25	11.8	5.3	68.3	46.9		32.36
New segmentation						
lm	10.9	9.8	73.1	54.1		34.93
lm-10	8.7	5.6	67.3	62.4		34.77

Table 8.1: Bleu scores obtained on ASR reference transcripts (Ref) as well as on ASR hypotheses (ASR) together with the average (avg) segment length and standard deviation (sdev). Precision (PRC) and Recall (RCL) of segmentation boundaries to punctuation marks are given as well.

	avg	sdev	Bleu	Bleu
			l043	t036+t037
sent	27.1	20.2	17.7	19.9
punct	11.1	9.6	18.1	19.9
lm-10	10.2	6.0	18.0	20.2
lm-8	8.9	4.4	18.0	19.9

Table 8.2: Comparison of the language model based segmentation (*lm-10*, *lm-8*) with segmentations at sentence boundaries (*sent*)and punctuation marks (*punct*) for English-to-Spanish lecture translation on *lectEval*. All results are obtained on ASR hypotheses.

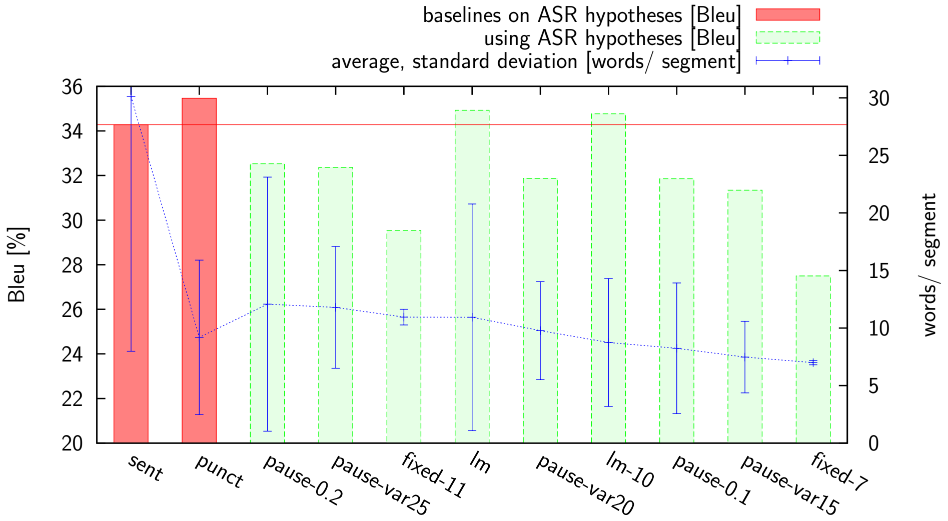


Figure 8.1: Results obtained by chunking ASR hypotheses sorted descending according to their average segment length. Bleu scores (left axis) are represented by boxes, the markers in the middle of the boxes give the average segment length (right axis) together with the standard deviation.

above-developed language model-based segmentation. As can be seen, the average segment length as well as the standard deviation can be significantly reduced with the help of the language model-based segmentation (*lm-10*), without any degradation in translation performance. Furthermore, by restricting the thresholds after eight words to reduce the latency further, only a slight degradation in translation quality was observed. It should be noted that all results were obtained on ASR hypotheses produced with the system described in Section 7.5 using 1 second intervals for partial trace-back. This system achieved a WER of 16.8% on *t036*, 13.7% on *t037*, and 16.0% on *l043*.

Chapter 9

A System for Simultaneous Speech Translation

Given the description of the individual components of the simultaneous translation system in the previous chapters, this chapter introduces the architecture of the simultaneous translation system implemented in this thesis. Figure 9.1 shows the system as deployed in a real lecture scenario.

9.1 Architecture and Information Flow

Figure 9.2 shows a slightly extended version of the schematic overview of the simultaneous translation system which was given in Chapter 3. The speech of the lecturer is recorded with the help of a close-talk microphone and processed by the speech recognition component. The partial hypotheses produced by this component are collected in the resegmentation component, for merging and re-splitting at appropriate “semantic” boundaries. The resegmented hypotheses are then transferred to one or more machine translation components, at least one per language pair. Different output technologies may be used for presenting the translations to the audience.

All components of the simultaneous translation system are connected together in a client-server framework, similar to the one described in [FWS$^+$01]. It is designed such that several consumers can process the output of one or more producers. Imagine for example a panel discussion instead of a lecture situation, where multiple people are able to speak at the same time. For such a situation, the architecture makes it possible to use several speech recognizers for processing the input signals. The different outputs are then ordered, merged and resegmented per speaker in different resegmentation components, and several machine translation systems can then be used to consume the output of the resegmentation components, and to translate it into the different required languages. For the identification of the different streams and for the merging and resegmentation process, unique speaker-

Figure 9.1: Simultaneous translation of a lecture with the system implemented in this thesis.

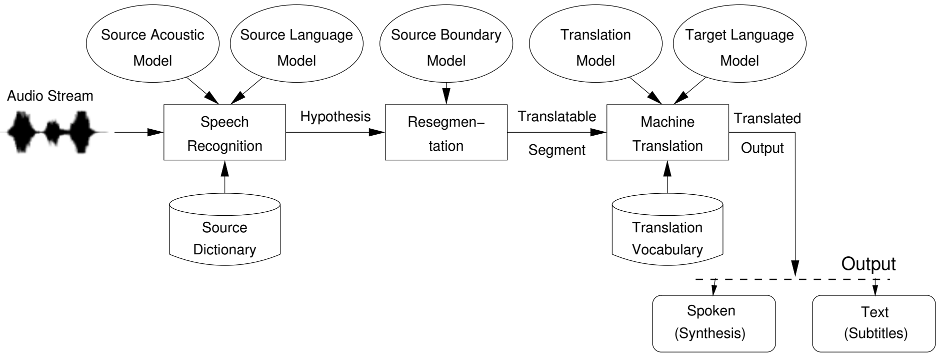


Figure 9.2: Schematic overview and information flow of the simultaneous translation system. The main components of the system are represented by cornered boxes and the models accessed by these components by ellipses. The different output forms are represented by rounded boxes.

and segment-related identifiers are used. The different translation streams are then multiplexed for delivery by the different output technologies. With this set-up, the overall latency of the system can be significantly reduced in such situations.

If required, the architecture also allows for combining the hypotheses of several ASR components given the same input signal in order to improve the recognition quality, by easily adding another consumer responsible for the combination. The only problem for the simultaneous translation system in this case is that the latency of the system not be affected.

9.2 Output Technologies

When offering simultaneous translation services, there is always the question on how to deliver the translation to an audience. In the case of human interpreters, the translation has always been delivered in spoken form, and typically headphones are used. An automatic system has the advantage that the translation can be delivered in written form also. In the following, the advantages and disadvantages of the different transmission forms, types, and output devices are discussed in more detail.

9.2.1 In Written Form

One of the most prominent transmission types is presenting the translation as subtitles on a projection screen. The advantages of this transmission type are that it is easy to implement, does not require any additional hardware and can be presented to several people at the same time. However, as the number of necessary output languages increases, the clarity of the presentation suffers. In contrast, with display devices assigned to a single person or a small group of persons of the same mother tongue, it is possible to deliver target-oriented translations in the required language only, but it is necessary to purchase additional hardware. The additional cost can be reduced significantly by also supporting standard handheld devices like PDAs or cellphones, which are becoming more and more widely used.

All above mentioned output devices have the major disadvantage that one is unable to look at the lecturer and presentation slides at the same time as reading the translations. For this reason, heads-up display goggles (see Figure 9.3) might be an ideal solution, because they enable the wearer to see both the lecturer and the translation at the same time. Problematic in this case is that equipping an audience with such a device is very expensive, and changing focus from the speaker to the translation device can become very tiring.

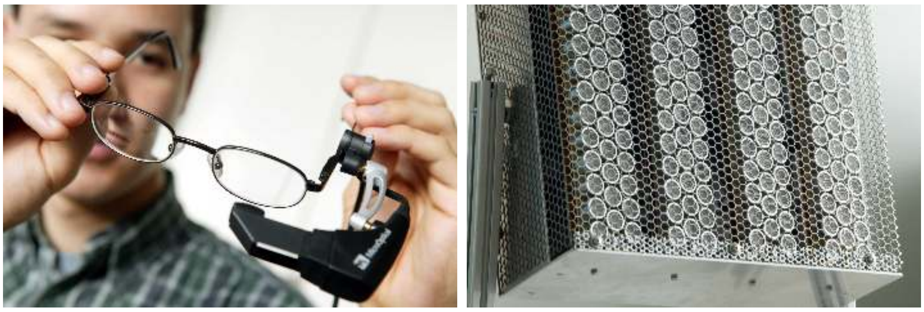


Figure 9.3: Different output devices: Translation Goggles on the left and the Targeted Audio device on the right.

9.2.2 In Spoken Form

Another prominent transmission form is delivering the translation with the help of a speech synthesizer in spoken form. Using standard loudspeakers for this is only suitable when the whole audience has the same mother tongue. Otherwise, the different spoken translations and original speech would interfere with each other. Using headphones solves this problem, but requires additional equipment for the audience. Furthermore, it hinders the audience from communicating with each other, because it is difficult to speak with a neighbor without taking off the headphone.

To address these problems, we have explored beam steered sound beams, which allow for delivering the translation to a desired group of people located in the beam using a single device only, without disturbing other people in the audience located outside of the beam [WF08]. With this solution, the communication between people in the audience is not hindered and the audience is not forced to wear additional personal devices. Furthermore, due to the fact that several people can be addresses with a single device, the additional equipment cost is lower.

For generating the steered sound beams, possible methods include using a sound transducer with a diameter much larger than the wavelength of the sound wave to be transmitted, or using standard array signal processing with multiple loudspeakers. Both have the disadvantage that the overall size of the device is large and thus unsuitable for a simultaneous lecture translation system. Therefore, in [OPL06, OL07], another solution was proposed which uses ultrasound to generate a narrow beam with an acceptable transducer dimension. The device uses the characteristic that, at high sound pressure levels, non-linear effects appear and the waveforms become distorted. This distortion can be calculated and used to advantage, such that the non-linearity of the air acts as a demodulator. The ultrasound wave is used as a carrier, which is modulated by an audio signal. Figure 9.3 shows such a targeted audio device. It consists of several small ultrasound loudspeakers

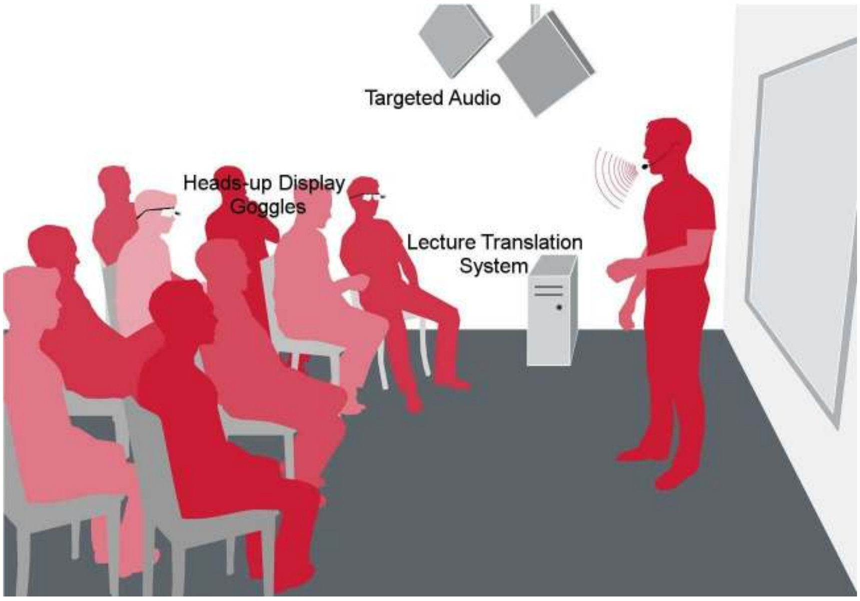


Figure 9.4: The lecture scenario.

and outputs audio in a beam of a width of about 1-2 meters.

9.3 Conclusion

Figure 9.4 shows again the lecture scenario already presented in Chapter 3. The lecturer is standing between the projection screen and the audience. The audio is processed, simultaneously translated by the lecture translation system and delivered to the audience with the help of targeted audio devices or heads-up display goggles. Several such targeted audio devices can be assigned to various languages to accommodate each participant in the lecture room.

Although several problems could be solved with the help of these devices, one problem still exists, namely the delivery rate problem. To keep the latency of the system constant, the system has to consume the input as fast as it is produced and therefore has to deliver the output with the same rate. This is especially problematic for language pairs for which an expression with the same meaning consists of more phonemes in the target language than in the source language. Figure 9.5 compares the average number of words in English necessary to phrase a single word of another language. It can be seen that e.g. a word in Finnish is phrased in almost 1.4 words in English and a word in French in almost 0.9 words in English. The results

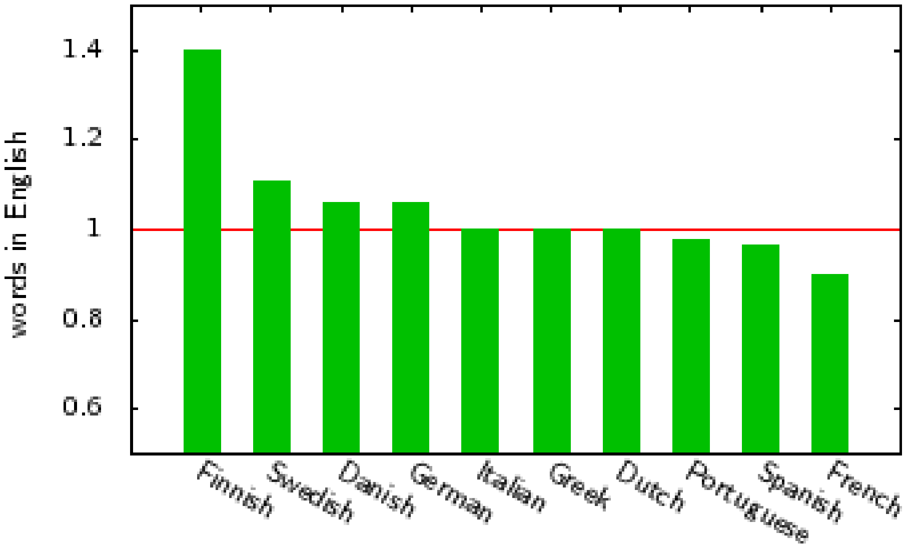


Figure 9.5: Comparison of the average number of words in English necessary to phrase a single word of another language measured on the Europarl [Koe05] corpus.

do not provide a direct comparison of the duration of a phrase in different languages, because words also differ in their durations; they are meant to provide a first impression. Furthermore, the automatic system is not perfect, and understanding an imperfectly written or spoken translation may not be an easy task. In such cases, increasing the delivery rate of the translation to keep up with the lecturer will make this task even more difficult.

Chapter 10

End-to-End Evaluation

By taking the results of the preliminary chapters into consideration, a real-time, low-latency, speaker and topic adapted English-to-Spanish simultaneous speech-to-speech translation system was constructed.

For speech recognition, the MMIE-trained acoustic model with 184M Gaussians over 3000 codebooks was adapted in a supervised manner on four hours of speech of the same speaker (*3000-64*). The *lectAdapt* language model was adapted using a three-fold linear interpolation of (1) a background language model, (2) a language model computed on general speaker-dependent topic data collected with the help of the speaker's past publications, and (3) a language model computed on specific topic data related to the current talk or lecture with the help of the presentation slides (*re-estimated S+Ltopic lectAdapt*). Where possible, the mixture weights were optimized on reference transcripts related to the current talk; otherwise, all reference transcripts in *lectOther* were used instead.

To reduce system latency, decoding was performed in one second intervals. Between two successive intervals, incremental VTLN and cMMLR adaptation were used and partial hypotheses were returned. To improve the real-time performance of the recognizer, Gaussian selection was applied which is based on clustering all Gaussians into 1024 clusters and selecting the top 64 clusters for each frame.

The partial hypotheses returned by the speech recognizer were resegmented using a language model and likelihood ratio thresholds (*lm-10*), and then transmitted to the machine translation component.

For machine translation, the translation model was adapted using a small number of manually translated lectures. For the target language model, a web data collection was performed by using topic- and speaking style-related n-grams as search queries. The language models computed using the web data as well as manually translated lectures were then interpolated with the existing background models.

In this Chapter, automatic evaluation results of the simultaneous trans-

Development Set		Evaluation Set	
	WER		WER
t035	12.4%	t036	12.9%
l003	11.5%	t037	10.3%
t012	13.1%	l043	13.2%
t032	10.5%	t038	14.8%
t041	12.1%	t044	13.6%
t042	9.0%		
Overall	11.9%		13.3%

Table 10.1: Detailed final automatic ASR evaluation results on the development and evaluation data.

	NIST	BLEU
t036+	5.60	23.0
l043	5.05	19.0

Table 10.2: Final automatic SMT evaluation results on the evaluation data. Only one reference translation was used for scoring the results.

lation system will be presented in Section 10.1. Human end-to-end evaluation results will be presented in Section 10.2 and compared in quality to an offline TC-STAR system for parliamentary speeches as well as a human interpreter.

10.1 Automatic End-to-End Evaluation

Table 10.1 shows the automatic ASR evaluation results. Overall, WERs of 11.9% and 13.3% were achieved on the development and the evaluation set, respectively. Decoding and adaptation on the partial hypotheses have a real-time factor of 0.93 and provides ample scope for variations in the speaking rate of the speaker. From the WERs per talk or lecture, it can be seen that *t038* is harder to recognize than the others. The explanation is traceable to the noisy environmental conditions during this talk.

In Table 10.2 shows the corresponding translation scores for the evaluation set.

10.2 Human Evaluation

The human end-to-end evaluation was carried out with the help of ELDA [1] as for the 2007 TC-STAR human end-to-end evaluation [HMC07]. To reduce

[1] Evaluations and Language resources Distribution Agency, `http://www.elda.org/index.php`

the overall effort and costs, six excerpts were used, three from each of the two talks *t036+* and *l043* of about 6 minutes each. The excerpts focus on different topics each and represent a total of about 36 minutes and 7,200 words. A professional human interpreter was asked to translate the same six excerpts, and the synthesized outputs of the automatic system and the human interpretation were then presented to human judges.

The evaluation was split into two parts. In the component evaluation, judges were able to directly compare the source excerpts and their translation. In the end-to-end evaluation, they were able to listen to the spoken translation of the excerpts only. It should be mentioned that the judges used for the first part of the evaluation were not used for the second part in order to prevent biased results. For both parts, the excerpts were evaluated in two categories: *fluency* and *adequacy*.

10.2.1 Component Evaluation

For this evaluation, 10 judges were recruited, of which 5 did not have specific knowledge of the domain of the lectures. All were Spanish native speakers. Each judge evaluated all six excerpts, and each segment was thus evaluated by 10 different judges. The judgments were made on a per-segment level, with the segmentation carried over from the manual transcriptions of the excerpts in the source language. As a result, segments could contain more than one sentence. There were about 247 segments in all.

Fluency was evaluated by judging the quality equated with the syntactical correctness of the segment. *Adequacy* referred to whether the meaning of a sentence in the source language was correctly transferred into the target language. In both cases, judgment was on a five-point scale, with a value of one for lowest and a value of five for highest quality.

Results

Table 10.3 shows the detailed automatic evaluation results of the translated output restricted to the excerpts. The overall word error rate of the ASR component was 11.9% with an OOV-rate of about 0.5%. On this output, the SMT achieved an overall BLEU score of about 28.94. Compared to the TC-STAR-07 system, it can be seen that there is a significant difference for both ASR and SMT performance. However, it should be kept in mind that (1) European Parliament Plenary Sessions are less difficult to recognize than lectures and (2) the simultaneous translation system runs in real-time, whereas the TC-STAR-07 translation system process the data offline in many times slower than real-time. Furthermore, it can be observed that *l043* is more difficult than *t036+*. This is due to the differences in topic between the two lectures, and lower coverage of the language and translation models.

	ASR	SMT	
	WER	BLEU	NIST
l043-1	14.6%	25.62	6.30
l043-2	14.5%	22.60	6.30
l043-3	9.6%	28.73	7.14
t036+1	11.4%	34.46	7.94
t036+2	12.1%	29.41	7.19
t036+3	9.2%	35.17	7.83
Overall	11.9%	28.94	7.07
TC-STAR-07	6.9%	40.6	9.19

Table 10.3: ASR and SMT automatic evaluation results restricted to the excerpts used for human evaluation. Two reference translations were used to evaluate MT performance. The last row shows the offline results of the final TC-STAR-07 system on different data.

	all judges	experts	non-experts
Fluency	3.13	2.84	3.42
Adequacy	3.26	3.21	3.31

Table 10.4: Averaged fluency and adequacy results for the human component evaluation.

Table 10.4 shows the averaged results for the adequacy and fluency judgments over all excerpts. Surprisingly, the results are lower for the experts than for the non-experts. A possible explanation could be that experts know the scientific vocabulary and are therefore more exigent. Another observation is that the adequacy score is higher than the fluency score, i.e. the simultaneous translation system is better in transferring the meaning than ensuring syntactic correctness.

10.2.2 End-to-End Evaluation

For this evaluation, 20 judges had to be recruited because of the additional evaluation of the human interpreter result. Again, the same demands were made on the judges as for the component evaluations, and each judge evaluated either all six automatic or all six human excerpts.

Since judges were able to listen to the audio only, fluency was evaluated by asking the judges to rate the overall quality of the current audio sample. It should be noted, that in contrast to the component evaluation, the quality of the human or synthetic speech was taken into account. The judgment was on a five-point scale, reaching from one for very bad and unusable up to 5 for very useful. Due to the lack of the source speech, a comprehension questionnaire was used for evaluating adequacy.

For each excerpt, the questionnaire consisted of 10 questions. Three types of questions were asked: simple factual (70%), yes/ no (20%), and list (10%). This was done in order to diversify the difficulty of the questions and to test the system on different levels. The questionnaires used for each excerpt can be found in Appendix A.

The excerpts were presented to the judges via an online interface together with the following evaluation instructions:

- listen to the excerpts only once
- answer the comprehension questionnaire
- judge the fluency
- go to the next excerpt

After evaluation, an objective verification or validation of the answers was performed by comparing the judges' answers with references. In order to identify where information was lost, the SMT, ASR and interpreter output was analyzed in more detail. For this purpose the following voting scheme was used:

0: the information or answer is totally missing

1: the information or answer is partially correct, but clearly incompleted

2: the information or answer is really close, but does not correspond exactly to the reference

3: the information or answer is undeniably correct

For further processing, the values where mapped to a binary scale, i.e. 0 and 1 was mapped to 0, and 2 and 3 were mapped to 1.

Results

Table 10.5 shows that while the fluency of the simultaneous translation system was judged to be slightly better than of the TC-STAR-07 system, the fluency of human interpreters is worse on the *Lectures* than on the EPPS task. This is most likely due to the more spontaneous speaking style of lectures and a higher familiarity of interpreters with the topics and the typical expressions used in the European Parliament.

Table 10.6 shows the percentage of good answers to the questionnaires given by judges from the output of interpreters and of the automatic systems. Both automatic systems perform worse than the interpreters, and the simultaneous translation system is worse than the TC-STAR system. This is clearly in accordance with the differences in WER and Bleu between the TC-STAR-07 and *Lecture* systems.

	interpreter	automatic system
TC-STAR-07	4.03	2.05
Lectures	3.03	2.35

Table 10.5: Human Evaluation, Overall Fluency (1 low - 5 better).

	interpreter	automatic system
TC-STAR-07	0.74	0.64
Lectures	0.74	0.52

Table 10.6: Human Evaluation, Overall Comprehension (0 low - 1 better).

In addition, an objective verification of the presence of the information needed for answering the questions was performed by a native Spanish speaker. Table 10.7 shows the percentage of answers present or the maximum number of answers found in the Spanish translation output and in the intermediate outputs of the ASR and SMT components. The comparison shows that for the lecture task, the human interpreter was able to retain 85% of the information needed to answer all questions, while the judges, who could listen to the interpreter output only once, then could answer 74% of the questions. The automatic system presented 58% of the necessary information to the judges who then answered 52% of the questions correctly; the ASR component loses 17% of the information, compared to 3% in the TC-STAR system.

To objectively compare the translation quality of the automatic system with the quality of the human interpreter, the results were limited to the questions for which the answers were included in the interpreter's speech as shown in Table 10.8. It can be seen that in this subset the interpreters are far better than they do over all data (from 0.74 to 0.80); this is also true to a smaller extent for the automatic systems (from 0.64 to 0.66 and from 0.52 to 0.53). However, as noted earlier there is already a loss of 20% in the ASR component.

	interpreter	automatic system	SMT	ASR
TC-STAR-07	0.91	0.89	0.92	0.97
Lectures	0.85	0.58	0.65	0.83

Table 10.7: Human Evaluation, Objective Comprehension (0 low - 1 better).

	interpreter	automatic system	TTS	SMT	ASR
TC-STAR-07	0.80	0.66	0.91	0.93	0.97
Lectures	0.80	0.53	0.60	0.70	0.80

Table 10.8: Human Evaluation, Limited Comprehension (0 low - 1 better).

Chapter 11

Conclusions

This thesis presented the first available prototype of a simultaneous speech-to-speech translation system particularly suited for lectures, speeches and other talks. It demonstrated how such a complex system can be build as well as the limitations of the current state-of-the-art. It compares and studies different technologies to meet the given constraints in real-time, latency as well as translation quality. With the help of this thesis one should be able to build such a system and to make an informed analysis of anticipated performance versus cost of the techniques presented.

The proposed simultaneous translation system consists of two main components, the automatic speech recognition and the statistical machine translation component. To meet the given constraints in latency and real-time without a drop in translation quality, several optimizations are necessary. The most obvious is to use adaptation. Within the proposed adaptation framework it is possible to apply adaptation on different levels, depending on the type of information available. Different speaker and topic adaptation techniques were studied with respect to the type and amount of data on hand. To reduce the latency of the system, different speed-up techniques were investigated. The interface between speech recognition and machine translation components was optimized to meet the given latency constraints with the help of a separate resegmentation component.

In the automatic end-to-end evaluation, the system showed an overall word error rate of 11.9% and a BLEU score of 28.94 on the six excerpts used for human evaluation. This is still rather low compared to an offline system for translating European parliament speeches (*TC-STAR-07*). However the offline system had no latency constraints, and parliament speeches are much easier to recognize and translate than compared to the more spontaneous lectures on which this thesis focuses. This clearly shows the difficulty of the whole task. However, the human end-to-end evaluation of the system in which the system is compared with human interpretation shows that the current translation quality allows for understanding of at least half of the content of a talk.

11.1 Thesis Results

In this thesis, different technologies required for implementing a simultaneous translation system are analyzed. The experiments presented in this thesis show that a satisfactory simultaneous translation system can be developed with current state-of-the-art technologies. Specifically, the following results, summarized in Table 11.1 for speech recognition, Table 11.2 for machine translation, and Table 11.3 for the resegmentation component, were achieved. Although word error rates are given for *lectDev* as well, the word error rates, real-time factors and relative improvements presented in the text below refer to *lectEval*.

Latency and Real-Time: An extensive study of the pruning parameters available in the Ibis decoder was carried out, which gave valuable insights on the influence of these parameters on recognition accuracy and decoding speed. Changes in the *state beam* have the larges impact on the ratio between recognition accuracy and decoding speed, followed by the *word beam*. The influence of the topN pruning parameters is small.

In addition, it was shown that with an improved implementation, Gaussian clustering has a better WER-to-RTF ratio than Gaussian selection using the Bucket Box Intersection algorithm. This is especially true if discriminative training was used for the acoustic model. Using these speed-ups, together with a smaller acoustic model, the real-time factor was reduced from 1.28 by 40% to 0.87 with only a minor increase in WER of 1.2% (Table 11.1).

To reduce the overall latency, the output of the speech recognition component was streamed. A partial trace-back mechanism was implemented with which partial hypotheses can be returned in short intervals of one or two seconds. The overall latency could be reduced from a few seconds to only about two seconds. Although the overhead increases with shorter intervals, the speech recognition component has a real-time factor of 0.93. This gives ample scope for variations in the speaking rate.

Speaker Adaptation: Speaker adaptation was successfully applied to the lecture translation system. The impact in WER was analyzed for different amounts of data, either in the form of reference or automatic transcripts. With 15 minutes of data – a common speaking time for presentations – the WER could be reduced by 6.5% if reference transcripts were used for updating the parameters of VTLN and MLLR. In

	lectDev	lectEval	
	WER	WER	total RTF
baseline 4000-64	15.1%	16.6%	1.28
Latency and Real-time			
+ AM optimization + speed-ups	15.4%	16.8%	0.77
+ reduced latency (1sec intervals)	15.2%	16.9%	0.93
Speaker Adaptation			
+ unsupervised AM adapt. (15min)	14.3%	16.0%	
+ supervised AM adapt. (15min)	14.3%	15.8%	
+ unsupervised AM adapt. (4hrs)	13.5%	15.3%	
+ supervised AM adapt. (4hrs)	12.8%	14.5%	
Topic Adaptation			
unsup. AM + lectBase (15min)	14.3%	16.0%	
+ re-est. S+Ltopic	14.0%	15.5%	
sup. AM + lectBase (15min)	14.4%	15.6%	
+ re-est. S+Ltopic	13.8%	15.1%	
unsup. AM + lectBase (4hrs)	13.4%	14.9%	
+ re-est. S+Ltopic	13.0%	14.6%	
sup. AM + lectBase (4hrs)	12.7%	14.0%	
+ re-est. S+Ltopic	12.1%	13.7%	
unsup. AM + lectAdapt (15min)	13.6%	15.2%	
+ re-est. S+Ltopic	13.7%	15.1%	
sup. AM + lectAdapt (15min)	13.6%	15.0%	
+ re-est. S+Ltopic	13.5%	14.7%	
unsup. AM + lectAdapt (4hrs)	12.8%	14.4%	
+ re-est. S+Ltopic	12.8%	14.3%	
sup. AM + lectAdapt (4hrs)	12.1%	13.6%	
+ re-est. S+Ltopic	11.9%	13.3%	

Table 11.1: Summarization of different techniques applied to the speech recognizer in order to reduce the latency and real-time and to increase the recognition quality. In rows labeled with *unsupervised* (unsup.), ASR hypotheses were used for offline acoustic model adaptation as well as for optimizing the language model mixture coefficients. In rows labeled with *supervised* (sup.), reference transcripts were used instead. The amount of data used for adaptation is given in parentheses.

	Unadapted		TM-adapt		+LM-adapt	
	NIST	BLEU	NIST	BLEU	NIST	BLEU
t036+, text input	5.72	23.4	5.89	25.0	6.33	28.4
t036+, ASR input	5.06	17.9	5.15	18.9	5.60	23.0
l043, text input	5.27	19.6	5.35	20.3	5.48	21.6
l043, ASR input	4.80	16.6	4.85	16.9	5.05	19.0

Table 11.2: Summarization of the SMT results obtained by language model and translation model adaptation.

	avg	sdev	Bleu	Bleu
			l043	t036+
sent	27.1	20.2	17.7	19.9
punct	11.1	9.6	18.1	19.9
lm-10	10.2	6.0	18.0	20.2
lm-8	8.9	4.4	18.0	19.9

Table 11.3: Summarization of the SMT results for different resegmentation approaches obtained on ASR hypotheses. In addition the average length of a segment (*avg*) together with the standard deviation (*sdev*) is shown. The results when hypotheses are manually split at sentence boundaries (*sent*) or punctuation marks (*punct*) are compared with an automatic language model based resegmentation algorithm (*lm-10, lm-8*).

the presence of four hours of data, the speed-optimized system could even be improved by 14.2% (from 16.9% to 14.5%). For the simultaneous translation system, the unsupervised adaptation results are of particular interest. The reason for this is that unsupervised adaptation can be performed automatically by using hypotheses of past talks from the translation system itself. The WER difference when using four hours of speech of the same speaker is 0.7% absolute, and 0.2% absolute for only 15 minutes of speech.

It was shown that even an acoustic model adapted using supervised methods can be further improved by using additional online adaptation.

A speaker clustering algorithm was presented, which is based on the cMLLR parameters used during system training, and with which the adaptation parameters for online cMLLR and VTLN adaptation were successfully initialized. It was shown that the recognition accuracy can be significantly improved on the first few utterances of a given speaker, especially for speaker-adaptively trained acoustic models.

Topic Adaptation: An adaptation framework for lectures and speeches was presented, which allows for a fully automatic adaptation of the simultaneous translation system towards specific topics of new presentations. It was shown that queries generated using a *tf-idf* based heuristic, can be successfully used to collect web data related to a specific talk or lecture. The impact of general data, collected with queries extracted from several publications of the speaker, was compared to a more specific collection with queries extracted from presentation slides. It was shown that both corpora are important and that improvements in WER up to 4-5% relative can be expected by using both, depending on the background language model (Table 6.16).

When other lectures of the same speaker become available, they can be used for a more specific optimization of the language model mixture coefficients as well as for background language model optimization. With the proposed adaptation framework, even a highly adapted language model (*lectAdapt*) can be improved by 2.2% in WER. Supervised adaptation was compared with unsupervised adaptation. When using with the same acoustic model, the difference in WER is at most 1.3% (see Table 6.16). When used together with an acoustic model adapted using unsupervised methods, the overall improvement in WER for the best-performing systems is 6.2% relative with *lectBase* and 7% relative with *lectAdapt*. Overall, the best word error rate of 13.3% was achieved with a speech recognizer adapted using supervised methods, with *lectAdapt* as a background language model.

The problem of vocabulary adaptation by adding new words extracted from the speakers publications or from presentation slides was also addressed. Although the OOV-Rate was significantly reduced by 35% relative, these improvements did not materialize in the WER (Table 6.16).

For machine translation, a system trained mainly on European Parliament speeches was used and successfully adapted towards the lecture domain. First, the translation model was adapted by using other lecture data and by adding hand-crafted common expressions, yielding an improvement of 2%-5% in BLEU. After this, the developed *tf-idf*-based web data collection was applied to the target language model. By using the collected data in addition to other lecture data, the translation score (BLEU) was improved by an additional 12%-21% (Table 11.2).

Resegmentation: To prepare and optimize the stream of words returned by the speech recognition component for machine translation, a resegmentation component was developed which tries to find semantic boundaries with the help of a statistical heuristic. It was shown that with this heuristic, using likelihood ratio thresholds on n-grams, the average segment length for English-Spanish translation systems can be kept around nine words with no increase in translation quality (Table 11.3). Nine words typically correspond to an ear-voice-span of around 4.5 seconds, and is similar to the ear-voice-span achieved by human interpreters for the investigated language pair English-Spanish.

Translation Delivery: Different ways were explored on how text or speech translation can be delivered to the audience. While head-phones are the most commonly used technique, innovative devices like targeted ultra-sound loudspeakers offer interesting alternatives.

Human Evaluation: Human evaluation of the system showed that, in comparison with the *TC-STAR-07* offline system evaluated on parliament speeches and the translation results of a human interpreter, the system implemented in this thesis performs worse, but still quite well. In fluency, the simultaneous translation system was evaluated better than the *TC-STAR-07* system, but worse than a human interpreter (2.35 vs. 3.03) (see Table 11.4). However, even the latter is only average on the used scale of 1-5. The reason for this is the technical nature and spontaneity of the lectures and talks. In terms of comprehension, the simultaneous translation system was judged to be 53%, which is smaller than the 66% of the *TC-STAR-07* system. The human interpreters achieved 80%. It was shown that more than half of the questions could be answered correctly. When the results of the simultaneous translation system are analyzed in more detail,

	interpreter	automatic system	TTS	SLT	ASR
Overall Fluency (1 low - 5 better)					
TC-STAR-07	4.03	2.05			
Lectures	3.03	2.35			
Limited Comprehension (0 low - 1 better)					
TC-STAR-07	0.80	0.66	0.91	0.93	0.97
Lectures	0.80	0.53	0.60	0.70	0.80

Table 11.4: Summary of the human evaluation results.

20% of the errors are contributed by the speech recognition component, an additional 10% by machine translation, and a further 10% to problems in speech synthesis (Table 11.4). Note that the development and optimization of a speech synthesis component was not part of this thesis.

Overall, the WER could be improved by almost 20% for an ASR system adapted using supervised methods compared to the baseline. For a speech recognizer adapted using unsupervised methods, the improvement was almost 14%. At the same time, the real-time factor was reduced by 27% relative. The improvement in translation quality was about 17% on average, compared to an unadapted system. The resegmented partial hypotheses of the best-performing ASR were used as SMT input.

11.2 Thesis Contribution

The experiments presented in this thesis lead to the following conclusions:

- With the final system, it is possible to understand at least half of the content of a simultaneously translated presentation. For a listener who is not able to understand the language of the speaker at all, this is quite helpful.

- The developed adaptation framework allows the system to automatically adapt to a specific speaker or topic. Performance improves as speakers continue using the system. Manually added information, such as publications, special terms and expressions, or transcripts or manuscripts of the presentation, definitely improve adaptation performance. The performance difference between automatically and manually generated information is larger when more data is available.

- The adaptation framework reduces the amount of time necessary for tailoring the lecture translation system towards a specific domain. A general domain-dependent language model can be adapted with the help of the adaptation schema implemented in this thesis, such that it performs similarly to a highly adapted language model using huge amounts of additional data.

- With the help of the studies about different adaptation techniques, speed-up techniques, and techniques for reducing the latency of the system, one can make an informed analysis of anticipated performance versus cost of the techniques presented.

- With respect to a human interpreter, the automatic system has the advantage that once it is adapted, it can be re-used relative cheaply.

- Interpreting is a very complex task for humans, and it is recommended to exchange interpreters at least every half an hour. Therefore, the simultaneous translation system is especially suitable for longer presentations such as lectures, or situations which are stressful for humans, such as environments with high background noise.

- The automatic system has no memory limitations. This means that for speakers with a high speaking rate, or when complicated sentence structures are used, the automatic system can be advantageous over a human interpreter. In such situations, the automatic system will definitely not drop information, instead, the latency will increase.

- The developed client-server framework allows to easily add new producers or consumers and therefore allows to easily tailor the system to the needs of different applications. For eaxample, multiple translations can be produced at the same time by just connecting several different translation components.

At the current time, automatic simultaneous translation is not used because it yields lower translation quality than a human interpreter. However, adoption depends on the cost-benefit ratio. In the authors opinion, in some situations such as small conferences or at universities such a system may be useful at the current performance level.

A first version of the system was presented in a press conference in October 2005 as well as at Interspeech 2006, where it was awarded as the "Best Presentation".

11.3 Recommendations for Future Work

The present work represents the first available prototype of a simultaneous translation system. Several aspects of the system have been analyzed and discussed, and the available prototype reveals new problems and allows for further studies.

- In the author's opinion, the most important improvement of the simultaneous translation system would be to port the streaming approach used in speech recognition to machine translation. Although segmentation in the form of inserting punctuation marks would still be necessary to increase the understandability of the output, a separate resegmentation component would be superfluous.

  One possibility for achieving this is to use systems based on finite state transducers (FST) such as [SBH06]. The theory of FSTs allows to merge the different systems into a single one so that global optimizations with respect to the overall translation quality, real-time behavior and latency can be applied.

- As seen in the human evaluation results, the performance of the ASR system is one bottleneck of the simultaneous translation system. To improve ASR performance common techniques such as lattice rescoring and consensus decoding should be ported to work in the streaming framework. The result of the ASR should be annotated with confidences to allow for more flexibility in translation.

- The support of multilinguality is another aspect which should be addressed. Especially in technical talks, many expressions may be in English while the rest of the talk is given in another language. A multilingual acoustic model is indispensable for recognizing multilingual expressions.

- Technical terms cannot be recognized if the corresponding words are not included in the vocabulary. Although the developed heuristic can significantly reduce the OOV-Rate, the words are not added in an optimal manner. The vocabulary adaptation can be improved to allow for the addition or exchange of even a larger amount of new words.

  In this context, it may be a good idea to allow for adaptation of the machine translation component. New expressions may sometimes be incorrectly translated with the existing system.

- Currently, the stream of words returned by the speech-recognition component is translated word-by-word, and only some disfluencies are filtered out. To improve the translation quality, an additional summarization component prior to translation may be a good idea, which also removes more complicate disfluencies and simplifies complicated sentence structures. Moreover, depending on the summarization level, the latency of the system can also be reduced.

- Unfortunately, an alignment of slides to the recorded presentations was not available. Motivated by results published by others, the author believes that this information can improve overall system performance. The recognition of words and expressions found on slides can be boosted to achieve a better recognition and translation.

- The presented output devices show interesting possibilities in delivering spoken translation. Further studies and development are necessary until these devices can exploit their full potential.

Bibliography

[AHH96] F. Alleva, X. Huang, and M.-Y. Hwang. Improvements of the Pronunciation Prefix Tree Search Organization. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Atlanta, USA, 1996.

[AKC94] Andreas Andreou, Terri Kamm, and Jordan Cohen. Experiments in vocal tract normalization. In *Proc. of the CAIP Workshop: Frontiers in Speech Recognition II*, 1994.

[AKESH00] Rajai Al-Khanji, Said El-Shiyab, and Riyadh Hussein. On the Use of Compensatory Strategies in Simultaneous Interpretation. *Meta : Journal des traducteurs*, 45(3):544–557, 2000.

[AKMS94] Tasos Anastasakos, Francis Kubala, John Makhoul, and Richard Schwartz. Adaptation to new microphones using tied-mixture normalization. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 433–436, Adelaide, Australia, 1994. IEEE.

[AMSM96] Tasos Anastasakos, John McDonough, Richard Schwartz, and John Makhoul. A compact model for speaker-adaptive training. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, pages 1137–1140, Philadelphia, PA, USA, 1996. ISCA.

[Aub99] X. Aubert. One Pass Cross Word Decoding for Large Vocabularies based on a Lexical Tree Search Organization. In *Proc. of the European Conference on Speech Communication and Technology (EUROSPEECH)*, Budapest, Hungary, 1999.

[Avi02] Edward Avis. Respell Version 0.1. http://membled.com/work/apps/respell/, 2002.

[Bar69] H. C. Barik. *A Study of Simultaneous Interpretation.* PhD thesis, University of North Carolina at Chapel Hill, 1969.

[BBFK05] K. Bain, S. Basson, A. Faisman, and D. Kanevsky. Accessibility, transcription, and access everywhere. *IBM Systems Journal*, 44(3):589–603, 2005.

[Bel00] Jerome R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *IEEE Transactions on Acoustics, Speech, and Signal Porcessing*, 88(8):63–75, August 2000.

[Bel04] Jerome R. Bellegarda. Statistical Language Model Adaptation: Review and Persepectives. *Speech Communication*, 42:93–108, 2004.

[Ben75] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the Association for Computing Machinery*, 18(9):509–517, Sep 1975.

[Boc93] E. Bocchieri. Vector Quantization for the effizient computation of continuous density likelihood. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 692–694, Minneapolis, USA, 1993. IEEE.

[BOS03] I. Bulyko, M. Ostendorf, and A. Stolcke. Getting more Mileage from Web Text Sources for Conversational Speech Language Modeling using Class-Dependent Mixtures. In *Proc. HLT-NAACL*, 2003.

[BT97] A. W. Black and P. A. Taylor. The Festival Speech Synthesis System: System documentation. Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Edinburgh, Scotland, United Kongdom, 1997. http://www.cstr.ed.ac.uk/projects/festival.

[CAB$^+$05] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The AMI meeting corpus: A Pre-Announcement. In *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005*, pages 28–39, Edinburgh, UK, 2005.

[CBF04] M. Cettolo, F. Brugnara, and M. Federico. Advances in the Automatic Transcription of Lectures. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada, 2004.

[CF98] M. Cettolo and D. Falavigna. Automatic Detection of Semantic Boundaries based on Acoutic and Lexical Knowledge. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, Sidney, Australia, 1998.

[CF06] M. Cettolo and M. Federico. Text Segmentation Criteria for Statistical Machine Translation. In *FinTAL – 5th International Conference on Natural Language Processing*, LNCS, pages 664 – 673, Turku, Finland, 2006. Springer Verlag Berlin/ Heidelberg.

[CG98] S. F. Chen and J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA, USA, 1998.

[Che99] C. Chen. Speech Recognition wiht Automatic Punctuation. In *Proc. of the European Conference on Speech Communication and Technology (EUROSPEECH)*, Budapest, Hungary, 1999.

[CMU] The CMU Pronunciation Dictionary v0.6. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[Cona] Linguistic Data Consortium. English Broadcast News Speech. http://www.ldc.upenn.edu. catalog IDs LDC97S44, LDC98S71.

[Conb] Linguistic Data Consortium. English Gigaword. http://www.ldc.upenn.edu. catalog ID LDC2003T05.

[Conc] Linguistic Data Consortium. Hansard French/English. http://www.ldc.upenn.edu. catalog ID LDC95T20.

[Cond] Linguistic Data Consortium. Translanguage English Database (TED) Speech. http://www.ldc.upenn.edu. catalog ID LDC2002S04.

[Cone] Linguistic Data Consortium. UN Parallel Text. http://www.ldc.upenn.edu. catalog ID LDC94T4A.

[Con04] Linguistic Data Consortium. ICSI, ISL and NIST Meeting Speech Coprora at LDC. http://www.ldc.upenn.edu, 2004. catalog IDs LDC2004S02, LDC2004S05, LDC2004S09.

[CSMR04] Arthur Chan, Jahanzeb Sherwani, Ravishankar Mosur, and Alex Rudnicky. Four-Layer Categorization Scheme of Fast

GMM Computation Techniques in Large Vocabulary Continuous Speech Recognition Systems. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, pages 689–692, Jeju Island, Korea, 2004. ISCA.

[Dir07a] Directorate-General for Interpretation. What we do – FAQ. http://scic.cec.eu.int/europa/display.jsp?id=c_5204, 2007. Online accessed 20-June-2007.

[Dir07b] Directorate-General for Translation. Translating for a Multilingual Community. http://ec.europa.eu/dgs/translation/bookshelf/brochure_en.pdf, 2007. Online accessed 20-June-2007.

[Dir07c] Directorate-General for Translation. Translation Tools and Workflow. http://ec.europa.eu/dgs/translation/bookshelf/tools_and_workflow_en.pdf, 2007. Online accessed 20-June-2007.

[Dod02] George Doddington. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurence Statistics. Technical report, NIST, 2002. http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf.

[DS05] Takao Doi and Eiichiro Sumita. Splitting Input for Machine Translation Using N-gram Language Model together with Utterance Similarity. *IEICE Transactions on Information and Systems*, 88(6):1256–1264, June 2005.

[Eur] The European Parliament Online. http://www.europarl.europa.eu.

[FA07] Jonathan Fiscus and Jerome Ajot. The Rich Transcription 2007 Speech-to-Text (STT) and Speaker Attributed STT (SASTT) Results. In *Rich Transcription 2007 Meeting Recognition Workshop*, Baltimore, MD, USA, May 2007. NIST. http://www.nist.gov/speech/tests/rt/2007/workshop/RT07-STT-v8.pdf.

[Fan60] Gunnar Fant. *Acoustic Theory of Speech Production*. Mouton & Co., Den Haag, Netherlands, 1960.

[Fed99] Marcello Federico. Efficient Language Model Adaptation through MDI Estimation. In *Proc. of the European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1583–1586, Budapest, Hungary, September 1999. ISCA.

[FGH+97] Michael Finke, Petra Geutner, Hermann Hild, Thomas Kemp, Klaus Ri es, and Martin Westphal. The Karlsruhe-VERBMOBIL Speech Recognition Engine. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Munich, Germany, 1997.

[FIK+06] Christian Fügen, Shajith Ikbal, Florian Kraft, Kenichi Kumatani, Kornel Laskowski, John W. McDonough, Mari Ostendorf, Sebastian Stüker, and Matthias Wölfel. The ISL RT-06S Speech-to-Text System. In Steve Renals, Samy Bengio, and Jonathan Fiskus, editors, *Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006, Bethesda, MD, USA*, volume 4299 of *Lecture Notes in Computer Science*, pages 407–418. Springer Verlag Berlin/ Heidelberg, 2006.

[Fis97] Jonathan G. Fiscus. A post-processing system to yield reduced word error rates: Recogniser Output Voting Error Reduction (ROVER). In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 347–354, Santa Barbara, California, USA, December 1997. IEEE.

[FK07] C. Fügen and M. Kolss. The Influence of Utterance Chunking on Machine Translation Performance. In *Proc. of the European Conference on Speech Communication and Technology (INTERSPEECH)*, Antwerp, Belgium, August 2007. ISCA.

[FKB+06] C. Fügen, M. Kolss, D. Bernreuther, M. Paulik, S. Stüker, S. Vogel, and A. Waibel. Open Domain Speech Recognition & Translation: Lectures and Speeches. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, 2006.

[FKFW99] M. Finke, D. Koll, J. Fritsch, and A. Waibel. Modeling and Efficient Decoding of Large Vocabulary Conversational Speech. In *Proceedings of the Eurospeech*, Budapest, Hungary, 1999.

[FKJ06] George Foster, Roland Kuhn, and Howard Johnson. Phrasetable Smoothing for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006.

[FKPW06] C. Fügen, M. Kolss, M. Paulik, and A. Waibel. Open Domain Speech Translation: From Seminars and Speeches to Lectures. In *TC-Star Speech to Speech Translation Workshop*, Barcelona, Spain, June 2006.

[FR96] Jürgen Fritsch and Ivica Rogina. The Bucket Box Intersection (BBI) Algorithm for Fast Approximative Evaluation of Diagonal Mixture Gaussians. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 837–840, Atlanta, GA, USA, May 1996. IEEE.

[FSS$^+$03] Christian Fügen, Sebastian Stüker, Hagen Soltau, Florian Metze, and Tanja Schultz. Efficient Handling of Multilingual Language Models. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 441–446, St. Thomas, Virgin Islands, USA, December 2003. IEEE.

[Fur05] Sadaoki Furui. Recent Progress in Corpus-Based Spontaneous Speech Recognition. *IEICE Transactions on Information and Systems*, E88-D(3):366–375, March 2005.

[Fur07] Sadaoki Furui. Recent advances in automatic speech summarization. In *Proc. Symposium on Large-Scale Knowledge Resources (LKR2007)*, pages 49–54, Tokyo, Japan, 2007.

[FWM$^+$06] C. Fügen, M. Wölfel, J. W. McDonough, S. Ikbal, F. Kraft, K. Laskowski, M. Ostendorf, S. Stüker, and K. Kumatani. Advances in Lecture Recognition: The ISL RT-06S Evaluation System. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, Pittsburgh, PA, USA, September 2006.

[FWS$^+$01] C. Fügen, M. Westphal, M. Schneider, T. Schultz, and A. Waibel. LingWear: A Mobile Tourist Information System. In *Proc. of the Human Language Technology Conf. (HLT)*, San Diego, California, March 2001. NIST.

[Gal97] M. J. F. Gales. Maximum Likelihood Linear Transformations for HMM-based Speech Recognition. Technical report, Cambridge University, Cambridge, United Kingdom, 1997.

[Gal98] M. J. F. Gales. Semi-tied Covariance Matrices. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1998.

[GAL05] GALE. Global Autonomous Language Exploitation. http://www.darpa.mil/ipto/programs/gale, 2005.

[GBK$^+$05] C. Gollan, M. Bisani, S. Kanthak, R. Schlüter, and H. Ney. Cross Domain Automatic Transcription on the TC-STAR EPPS Corpus. In *Proc. of the International Conference on*

Acoustics, Speech, and Signal Processing (ICASSP), Philadelphia, PA, USA, 2005.

[GFS98] P. Geutner, M. Finke, and P. Scheytt. Adaptive Vocabularies for Transcribing Multilingual Broadcast News. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seattle, Washington, USA, May 1998.

[GHC$^+$07] James Glass, Timothy J. Hazen, Scott Cyphers, Igor Malioutov, David Huynh, and Regina Barzilay. Recent Progress in the MIT Spoken Lecture Processing Project. In *Proc. of the European Conference on Speech Communication and Technology (INTERSPEECH)*, pages 2553–2556, Antwerp, Belgium, August 2007. ISCA.

[GHSN07] Christian Gollan, Stefan Hahn, Ralf Schlüter, and Hermann Ney. An Improved Method for Unsupervised Training of LVCSR Systems. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, pages 2101–2104, Antwerp, Belgium, 2007. ISCA.

[GL94] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observation of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, April 1994.

[GLA98] Jean-Luc Gauvain, Lori Lamel, and Gilles Adda. The LIMSI 1997 Hub-4E Transcription System. In *Proc. of the DARPA Broadcast News Transcription & Understanding Workshop*, pages 75–79, Landsdowne, VA, USA, February 1998.

[GLM$^+$04] John S. Garofolo, Christophe D. Laprun, Martial Michel, Vincent M. Stanford, and Elham Tabassi. The NIST Meeting Room Pilot Corpus. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, 2004.

[GS06] Dirk Gehrig and Thomas Schaaf. A Comparative Study of Gaussian Selection Methods in Large Vocabulary Continuous Speech Recognition. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, pages 625–628, Pittsburgh, PA, USA, 2006. ISCA.

[GWK02] R. E. Gorin, Pace Willisson, and Geoff Kuenning. Ispell. http://ficus-www.cs.ucla.edu/geoff/ispell.html, 2002. Version 3.3.02 12 Jun 2005.

[HAH01] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing.* Prentice Hall PTR, NJ, 2001.

[Hen71] P. V. Hendricks. *Simultaneous Interpreting: A Practical Book.* Longman, London, 1971.

[Hen82] J. A. Henderson. Some Psychological Aspects of Simultaneous Interpreting. *The Incorporated Linguist*, 21(4):149–150, 1982.

[HHP01] Timothy J. Hazen, I.Lee Hetherington, and Alex Park. FST-Based Recognition Techniques for Multi-Lingual and Multi-Domain Spontaneous Speech. In *Proc. of the European Conference on Speech Communication and Technology (EUROSPEECH)*, Aalborg, Denmark, September 2001.

[HMC07] O. Hamon, D. Mostefa, and K. Choukri. End-to-end evaluation of a speech-to-speech translation system in TC-STAR. In *Proc. of Machine Translation Summit XI*, pages 223–230, Copenhagen, Denmark, 10-14. September 2007.

[HS92] W. John Hutchins and Harold L. Somers. *An Introduction to Machine Translation.* Academic Press, San Diego, 1992.

[HWC$^+$06] Jing Huang, Martin Westphal, Stanley Chen, Olivier Siohan, Daniel Povey, Vit Libal, Alvaro Soneiro, Henrik Schulz, Thomas Ross, and Gerasimos Potamianos. The IBM Rich Transcription Spring 2006 Speech-to-Text System for Lecture Meetings. In Steve Renals, Samy Bengio, and Jonathan Fiskus, editors, *Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006, Bethesda, MD, USA*, volume 4299 of *Lecture Notes in Computer Science*, pages 432–443. Springer Verlag Berlin/ Heidelberg, 2006.

[HWM03] Takaaki Hori, Daniel Willett, and Yasuhiro Minami. Language Model Adaptation using WFST-based Speaking-Style Translation. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 228–231, Hong Kong, 2003. IEEE.

[HZ02] J. Huang and G. Zweig. Maximum Entropy Model for Punctuation Annotation from Speech. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, Denver, CO, USA, 2002.

[Hö97] H. Hönig. Using text mappings in teaching consecutive interpreting. In C. Hauenschild and S. Heizmann, editors, *Machine Translation and Translation Theory.* Mouton de Gruyter, 1997.

[Int08] Intel. Intel C++ Compiler Documentation. http://www.intel.com/software/products/compilers/docs/clin/main_cls/index.htm, 2008. Document number: 304967-021US, Online accessed 04-June-2008.

[IO99] R. Iyer and M. Ostendorf. Relevance Weighting for combining multi domain data fo n-gram language modeling. *Computer Speech and Language*, 13(3):267–282, 1999.

[JAB$^+$04] A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, and B. Wrede. The ICSI Meeting Project: Ressources and Research. In *Proc. of the ICASSP Meeting Recognition Workshop*, Montreal, Canada, May 2004. NIST.

[Jon98] R. Jones. *Conference Interpreting Explained*. St. Jerome Publishing, Manchester, 1998.

[KdM90] Roland Kuhn and Renato de Mori. A Cache-based Natural Language Method for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–582, June 1990.

[KFSW05] Thilo W. Köhler, Christian Fügen, Sebastian Stüker, and Alex Waibel. Rapid Porting of ASR-Systems to Mobile Devices. In *Proc. of the European Conference on Speech Communication and Technology (INTERSPEECH)*, pages 233–236, Lisbon, Portugal, 2005. ISCA.

[KGY96] K. Knill, M. Gales, and S. Young. Use of Gaussian Selection in Large Vocabulary Continuous Speech recogniiton Using HMMs. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, pages 470–473, Philadelphia, PA, USA, 1996. ISCA.

[KM06] Philipp Koehn and Christof Monz. Manual and Automatic Evaluation of Machine Translation between European Languages. In *Workshop on Statistical Machine Translation*, New York, USA, 2006.

[KNRM02] Stephan Kanthak, Hermann Ney, Michael Riley, and Mehryar Mohri. A Comparison of Two LVR Search Optimization Techniques. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, pages 1309–1312, Denver, CO, USA, 2002. ISCA.

[Koe05] Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *The 10th Machine Translation Summit (MT Summit X)*, Phuket, Thailand, September, 12-16 2005. Corpus available at http://www.statmt.org/europarl/.

[Kop94] A. Kopczynski. *Bridging the Gap: Empirical Research in Simultaneous Interpretation*, chapter Quality in Conference Interpreting: Some Pragmatic Problems, pages 87–100. John Benjamins, Amsterdam/ Philadelphia, 1994.

[KP02] Dietrich Klakow and Jochen Peters. Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1):19–28, 2002.

[KSN00] Stephan Kantak, Kai Schütz, and Herman Ney. Using SIMD Instructions for fast Likelihood Calculation in LVCSR. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1531–1534, Istanbul, Turkey, 2000. IEEE.

[Kur03] Ingrid Kurz. Physiological stress during simultaneous interpreting: a comparison of experts and novices. In *The Interpreters' Newsletter*, 12, pages 51–67. EUT - Edizioni Università di Trieste, 2003.

[KW01] J.-H. Kim and P. C. Woodland. The use of Prosody in a combined System for punctuation Generation and Speech Recognition. In *Proc. of the European Conference on Speech Communication and Technology (EUROSPEECH)*, Aalborg, Denmark, 2001.

[KZK00] Sung Dong Kim, Byoung-Tak Zhang, and Yung Taek Kim. Reducing Parsing Complexity by Intra-Sentence Segmentation based on Maximum Entropy Model. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 13, pages 164–171, Hong Kong, China, 2000.

[KZV$^+$06] M. Kolss, B. Zhao, S. Vogel, A. Venugopal, and Y. Zhang. The ISL Statistical Machine Translation System for the TC-STAR Spring 2006 Evaluations. In *TC-Star Workshop on Speech-to-Speech Translation*, Barcelona, Spain, December 2006.

[Kö04] Thilo Köhler. Geschwindigkeitsoptimierung eines Spracherkenners für Handheld PCs. Master's thesis, Universität Karlsruhe (TH), Karlsruhe, Germany, 2004. Supervisors Christian Fügen and Sebastian Stüker.

[LAOPR06] Y. Lee, Y. Al-Onaizan, K. Papineni, and S. Roukos. IBM Spoken Language Translation System. In *TC-Star Speech to Speech Translation Workshop*, Barcelona, Spain, 2006.

[LBA$^+$06] Lori Lamel, Eric Bilinski, Gilles Adda, Jean-Luc Gauvain, and Holger Schwenk. 457–468. In Steve Renals, Samy Bengio, and Jonathan Fiskus, editors, *Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006, Bethesda, MD, USA*, volume 4299 of *Lecture Notes in Computer Science*. Springer Verlag Berlin/ Heidelberg, 2006.

[LBG$^+$06] J. Lööf, M. Bisani, Ch. Gollan, G. Heigold, B. Hoffmeister, Ch. Plahl, R. Schlüter, and H. Ney. The 2006 RWTH Parliamentary Speeches Transcription System. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, pages 105–108, Pittsburgh, PA, USA, September 2006. ISCA.

[(LD] Linguistic Data Consortium (LDC). http://www.ldc.upenn.edu.

[Led78] M. Lederer. Simultaneous Interpretation: Units of Meaning and Other Features. In David Gerver and H. Wallace Sinaiko, editors, *Language Interpretation and Communication*, pages 323–332. Plenum Press, New York and London, 1978.

[LFS07] Kornel Laskowski, Christian Fügen, and Tanja Schultz. Simultaneous Multispeaker Segmentation for Automatic Meeting Recognition. In *Proc. of the 15th EURASIP European Signal Processing Conference (EUSIPCO2007)*, pages 1294–1298, Poznań, Poland, September 2007.

[LGA$^+$07] L. Lamel, J.-L. Gauvain, G. Adda, C. Barras, E. Bilinski, O. Galibert, A. Pujol, H. Schwenk, and X. Zhu. The LIMSI 2006 TC-STAR EPPS Transcription Systems. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 997–1000, Honolulu, Hawaii, April 2007. IEEE.

[LGCL96] Alon Lavie, Donna Gates, Noah Coccaro, and Lori Levin. Input Segmentation of Spontaneous Speech in JANUS: a Speech-to-Speech Translation System. In Elisabeth Maier, Marion Mast, and Susann LuperFoy, editors, *Workshop on Dialogue Processing in Spoken Language Systems*, volume 1236 of *Lecture Notes in Computer Science*, pages 86–99. Springer Verlag Berlin/ Heidelberg, 1996.

[LGLW98] L. Levin, D. Gates, A. Lavie, and A. Waibel. An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, pages 1155–1158, Sydney, Australia, November 1998. ISCA.

[Liu04] Y. Liu. *Structural Event Detection for Rich Transcription of Speech*. PhD thesis, Purdue University, 2004.

[Loe98] D. Loehr. Can Simultaneous Interpretation Help Machine Translation. In D. Farwell et al., editor, *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas, AMTA'98, Langhorne, PA, USA, October 1998*, volume 1529/1998 of *Lecture Notes in Computer Science*, pages 213–224. Springer Verlag Berlin/ Heidelberg, 1998.

[LSF$^+$94] L.F. Lamel, F. Schiel, A. Fourcin, J. Mariani, and H. Tillmann. The Translanguage English Database TED. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, volume LDC2002S04, Yokohama, September 1994. LDC.

[LW95] C. J. Leggetter and P. C. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer Speech and Language*, 9:171–185, 1995.

[Man01] Inderjeet Mani. *Automatic Summarization*, volume 3 of *Natural Language Processing*. John Benjamins Publishing Company, 2001.

[MBH99] M. Mahajan, D. Beeferman, and X.D. Huang. Improved topic-dependent language modeling using informationretrieval techniques. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 541–544, Phoenix, AZ, USA, 1999. IEEE.

[MBS99] L. Mangu, E. Brill, and A. Stolcke. Finding Consensus among Words: Lattice-based Word Error Minimization. In *Proc. of the European Conference on Speech Communication and Technology (EUROSPEECH)*, Budapest, Hungary, 1999.

[MKSW99] John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. Performance measures for information extraction. In *In Proceedings of DARPA Broadcast News Workshop*, pages 249–252, 1999.

[MLBN05] E. Matusov, G. Leusch, O. Bender, and H. Ney. Evaluating Machine Translation Output with Automatic Sentence Segmentation. In *International Workshop on Spoken Language Translation*, Pittsburgh, PA, USA, 2005.

[MM96] B. Moser-Mercer. Lecture. Georgetown University, 1996.

[MMKK98] Barbara Moser-Mercer, Alexander Kunzli, and Marina Korac. Prolonged turns in interpreting: Effects on quality, physiological and psychological stress (Pilot Study). *Interpreting: International journal of research and practice in interpreting*, 3(1):47–64, 1998.

[MMN06] E. Matusov, A. Mauser, and H. Ney. Automatic Sentence Segmentation and Punctuation Prediction for Spoken Language Translation. In *Internat. Workshop on Spoken Language Translation*, Kyoto, Japan, 2006.

[MOS06] Arindam Mandal, Mari Ostendorf, and Andreas Stolcke. Speaker Clustered Regression-Class Trees for MLLR-Adaptation. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, pages 1133–1136, Pittsburgh, PA, USA, 2006. ISCA.

[MPR00] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted Finite-State Transducers in Speech Recognition. In *ITRW Automatic Speech Recognition: Challenges for the Millenium*, pages 97–106, Paris, France, 2000. ISCA.

[MSJN97] Spyros Matsoukas, Rich Schwartz, Hubert Jin, and Long Nguyen. Practical Implementations of Speaker-Adaptive Training. In *Proc. of the DARPA Speech Recognition Workshop*, pages 133–138, Chantilly, VA, USA, February 1997.

[NHUTO92] H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder. Improvements in Beam Search for 10000-Word continuous Speech Recognition. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, San Francisco, USA, 1992.

[NIS04] NIST. NIST MT evaluation kit version 11a. http://www.nist.gov/speech/tests/mt, 2004.

[Och03] F. J. Och. Minimum error rate training in statistical machine translation. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July 2003.

[Ode95] J. Odell. *The Use of Context in Large Vocabulary Speech Recognition.* PhD thesis, University of Cambridge, United Kingdom, 1995.

[OEN98] S. Ortmanns, A. Eiden, and H. Ney. Improved Lexical Tree Search for Large Vocabulary Speech Recognition. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seattle, USA, 1998.

[OFN97] Stefan Ortmanns, Thorsten Firzlaff, and Hermann Ney. Fast Likelihood Computation Methods for Continuous Mixture Densities in Large Vocabulary Speech Recognition. In *Proc. of the European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 139–142, Rhodes, Greece, Sep 1997. ISCA.

[OL07] Dirk Olszewski and Klaus Linhard. Highly directional multi-beam audio loudspeaker. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, pages 2630–2633, Pittsburgh, PA, USA, September 2007.

[ON03] F.J. Och and H. Ney. A Systematic Comparison of Various Statistical Alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[ONE96] S. Ortmanns, H. Ney, and A. Eiden. Language-model look-ahead for large vocabulary speech recognition. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, pages 2095–2098, Philadelphia, PA, USA, 1996.

[OPL06] Dirk Olszewski, Fransiskus Prasetyo, and Klaus Linhard. Steerable Highly Directional Audio Beam Louspeaker. In *Proc. of the European Conference on Speech Communication and Technology (INTERSPEECH)*, Lisboa, Portugal, September 2006.

[Par78] H. McIlvaine Parsons. Human factors approach to simultaneous interpretations. In David Gerver and H. Wallace Sinaiko, editors, *The Court Management & Administration Report*, number 3.10-16 in Court Interpreting, pages 315–321. New York: Plenum, Michelsen, Patricia. 1992, 1978.

[Pov05] Daniel Povey. *Discriminative Training for Large Vocabulary Speech Recognition.* PhD thesis, Peterhouse College & CU Engineering Departement, 2005.

[PRWZ02] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176 (W0109-022), IBM Research Division, T. J. Watson Research Center, 2002.

[RHP03] B. Ramabhadran, J. Huang, and M. Picheny. Towards Automatic Transcription of Large Spoken Archives – English ASR for the Malach Project. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 216–219, Hong Kong, China, April 2003. IEEE.

[RLS07a] Sharath Rao, Ian Lane, and Tanja Schultz. Improving spoken language translation by automatic disfluency removal: evidence from conversational speech transcripts. In *Proc. of MT Summit XI*, pages 385–389, Copenhagen, Denmark, September 2007.

[RLS07b] Sharath Rao, Ian Lane, and Tanja Schultz. Optimizing Sentence Segmentation for Spoken Language Translation. In *Proc. of the European Conference on Speech Communication and Technology (INTERSPEECH)*, Antwerp, Belgium, 2007. ISCA.

[Rog97] Ivica Rogina. *Parameterraumoptimierung für Diktiersysteme mit unbeschränktem Vokabular*. PhD thesis, Universität Karlsruhe (TH), 1997.

[RS02] Ivica Rogina and Thomas Schaaf. Lecture and Presentation Tracking in an Intelligent Meeting Room. In *Fourth IEEE International Conference on Multimodal Interfaces (ICMI'02)*, pages 47–52, Pittsburgh, PA, USA, October 2002. IEEE.

[RSM$^+$06a] B. Ramabhadran, O. Siohan, L. Mangu, G. Zweig, M. Westphal, H. Schulz, and A. Soneiro. The IBM 2006 Speech Transcription System for European Parliamentary Speeches. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, pages 1225–1228, Pittsburgh, PA, USA, September 2006. ISCA.

[RSM$^+$06b] B. Ramabhadran, O. Siohan, L. Mangu, G. Zweig, M. Westphal, H. Schulz, and A. Soneiro. The IBM 2006 Speech Transcription System for European Parliamentary Speeches. In *TC-Star Speech to Speech Translation Workshop*, Barcelona, Spain, June 2006.

[SBH06] Stephan Kanthak Srinivas Bangalore and Patrick Haffner. Finite-State Transducer-based Statistical Machine Translation using Joint Probabilities. In *Proc. of the International Workshop of Spoken Translation (IWSLT)*, pages 16–22, Kyoto, Japan, 2006.

[SCER97] Kristie Seymore, Stanley Chen, Maxine Eskenazi, and Ronald Rosenfeld. Language and Pronunciation Modeling in the CMU 1996 Hub 4 Evaluation. In *Proc. of the DARPA Speech Recognition Workshop*, pages 141–146, Chantilly, Virginia, USA, February 1997.

[Sch04] Thomas Schaaf. *Erkennen und Lernen neuer Wörter.* PhD thesis, Universität Karlsruhe (TH), 2004.

[SDMW06] Frederic Stouten, Jacques Duchateau, Jien-Pierre Martens, and Patrick Wambacq. Coping with disfluencies in spontaneous speech recognition: Acoustic detection and linguistic context manipulation. *Speech Communiaction*, 48:1590–1606, 2006.

[SDS$^+$06] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proc. of the 7th Conf. of Association for Machine Translation in the Americas (AMTA)*, pages 223–231, Cambridge, MA, USA, August 2006.

[Sel78] Danica Seleskovitch. *Interpreting for International Conferences: Problems of Language and Communication.* Pen & Booth, Washington DC, 1978.

[SFH$^+$06] S. Stüker, C. Fügen, R. Hsiao, S. Ikbal, F. Kraft Q. Jin, M. Paulik, M. Raab, Y.-C. Tam, and M. Wölfel. The ISL TC-STAR Spring 2006 ASR Evaluation Systems. In *TC-Star Speech to Speech Translation Workshop*, Barcelona, Spain, June 2006.

[SGN05] Abhinav Sethy, Panayiotis G. Georgiou, and Shrikanth Narayanan. Building Topic Specific Language Models from Webdata using Competitive Models. In *Proc. of the European Conference on Speech Communication and Technology (INTERSPEECH)*, pages 1293–1296, Lisbon, Portugal, 2005. ISCA.

[Sha48] Claude E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[SK06] Tanja Schultz and Katrin Kirchhoff. *Multilingual Speech Processing.* Elsevier, Academic Press, 2006.

[SMFW01] Hagen Soltau, Florian Metze, Christian Fügen, and Alex Waibel. A One Pass-Decoder Based on Polymorphic Linguistic Context Assignment. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Trento, Italy, 2001.

[SMFW02] Hagen Soltau, Florian Metze, Christian Fügen, and Alex Waibel. Efficient Language Model Lookahead through Polymorphic Linguistic Context Assignment. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 709–712, Orlando, FL, USA, May 2002. IEEE.

[SPK+07] S. Stüker, M. Paulik, M. Kolss, C. Fügen, and A. Waibel. Speech Translation Enhanced ASR for European Parliament Speeches - On the Influence of ASR Performance on Speech Translation. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, 2007.

[SPZ05] George Saon, Daniel Povey, and Geoffrey Zweig. Anatomy of an extremely fast LVCSR decoder. In *Proc. of the European Conference on Speech Communication and Technology (INTERSPEECH)*, pages 549–552, Lisbon, Portugal, Sep 2005. ISCA.

[SS96] A. Stolcke and E. Shriberg. Statistical language modeling for speech disfluencies. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 405–408, Atlanta, USA, 1996.

[Sto98] Andreas Stolcke. Entropy-based Pruning of Backoff Language Models. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pages 270–274, Lansdowne, VA, USA, February 1998.

[Sto02] A. Stolcke. SRILM – An Extensible Language Modeling Toolkit. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, Colorado, USA, 2002. ISCA.

[SW01] Tanja Schultz and Alex Waibel. Language Independent and Language Adaptive Acoustic Modeling for Speech Recognition. *Speech Communication*, 35, August 2001.

[SYM+04] H. Soltau, H. Yu, F. Metze, C. Fügen, Q. Jin, and S.-C. Jou. The 2003 ISL Rich Transcription System for Conversational

Telephony Speech. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada, 2004. IEEE.

[SZK$^+$03] George Saon, Geoffrey Zweig, Brian Kingsburry, Lidia Mangu, and Upendra Chaudhari. An Architecture for Rapid Decoding of Large Vocabulary Conversational Speech. In *Proc. of the European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1977–1980, Geneva, Switzerland, 2003. ISCA.

[TNN06] Isabel Trancoso, Ricardo Nunes, and Luís Neves. Classroom Lecture Recognition. In *Computational Processing of the Portuguese Language: 7th International Workshop, PROPOR 2006, Itatiaia, Brazil*, volume 3960 of *Lecture Notes in Computer Science*, pages 190–199. Springer Verlag Berlin/ Heidelberg, 2006.

[Tru99] Arturo Trujillo. *Translation Engines: Techniques for Machine Translation.* Series on Applied Computing. Springer Verlag Berlin/ Heidelberg, 1999.

[TS04] TC-STAR. Technology and Corpora for Speech to Speech Translation. http://www.tc-star.org, 2004.

[Vid97] Mirta Vidal. New Study on Fatigue Confirms Need for Working in Teams. *Proteus: Newsletter of the National Association of Judiciary Interpreters and Translators*, VI(1), 1997.

[VNT96] S. Vogel, H. Ney, and C. Tillmann. HMM-based Word Alignment in Statistical Translation. In *COLING 96*, pages 836–841, Copenhagen, 1996.

[Vog03] S. Vogel. SMT Decoder Dissected: Word Reordering. In *Int. Conf. on Natural Language Processing and Knowledge Engineering (NLP-KE)*, Beijing, China, 2003.

[Vog05] S. Vogel. PESA: Phrase Pair Extraction as Sentence Splitting. In *Machine Translation Summit 2005*, Thailand, 2005.

[VS06] Accipio Consulting Volker Steinbiss. Sprachtechnologien für Europa. www.tc-star.org/pubblicazioni/D17_HLT_DE.pdf, April 2006.

[VW03] Anand Venkataraman and Wen Wang. Techniques for Effective Vocabulary Selection. In *Proc. of the European Conference on Speech Communication and Technology (INTERSPEECH)*, pages 245–248, Geneva, Switzerland, September 2003. ISCA.

[WB01] Matthias Wölfel and Susanne Burger. The ISL Baseline Lecture Transcription System for the TED Corpus. Technical report, Universität Karlsruhe (TH), Interactive Systems Laboratories, 2001. http://isl.ira.uka.de/~wolfel/TR0001.pdf.

[WF08] Alex Waibel and Christian Fügen. Spoken Language Translation. *IEEE Signal Processing Magazine*, May 2008.

[WH06] Vincent Wan and Thomas Hain. Strategies for Language Model Web-Data Collection. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1069–1072, Toulouse, France, 2006. IEEE.

[Wik07a] Wikipedia. European Parliament – Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Interpreting, 2007. Online accessed 20-June-2007.

[Wik07b] Wikipedia. Interpreting – Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Interpreting, 2007. Online accessed 20-June-2007.

[Wik07c] Wikipedia. Translation – Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Translation, 2007. Online accessed 20-June-2007.

[Wos98] M. Woszczyna. *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition.* PhD thesis, University of Karlsruhe, Germany, 1998.

[WSS04] A. Waibel, H. Steusloff, and R. Stiefelhagen. CHIL – Computers in the Human Interaction Loop. In *5th International Workshop on Image Analysis for Multimedia Interactive Services*, Lisbon, April 2004. http://chil.server.de.

[WSTI95] T. Watanabe, K Shinoda, K Takagi, and K. Iso. High Speed Speech Recognition Using Tree.Structured Probability Density Functions. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 556–559, Detroit, USA, 1995. IEEE.

[XZN05] Jia Xu, Richard Zens, and Hermann Ney. Sentence Segmentation Using IBM Word Alignment Model 1. In *Proc. of the 10th annual Conference of the European Association for Machine Translation (EAMT)*, pages 280–287, Budapest, May, Hungary 2005.

[Yag00] Sane M. Yagi. Studying Style in Simultaneous Interpretation. *Meta : Journal des traducteurs*, 45(3):520–547, 2000.

[YLC06] J. Yuan, M. Liberman, and C. Cieri. Towards an Integrated Understanding of Speaking Rate in Conversation. In *Proc. of the International Conference on Speech and Language Processing (INTERSPEECH)*, Pittsburgh, PA, USA, September 2006. ISCA.

Appendix A

Questionnaires

This chapter presents the questionnaires in Spanish and English used for the human end-to-end evaluation.

l043-1

```
PREGUNTAS:
01) ¿Cuál es el tema de la conferencia?
02) ¿Cómo denomina el disertante al aparato humano que produce el discurso
    oral?
03) De acuerdo con el disertante, si tuviésemos que inventar una nueva
    estrategia de codificación para comunicarnos entre nosotros,
    ¿sería idéntica a la que hemos utilizado hasta este momento?
04) ¿Cuáles son las tres partes que componen el aparato humano que produce
    el discurso oral?
05) ¿Con qué compara el disertante al funcionamiento de la laringe?
06) ¿De qué es responsable la laringe?
07) En lugar de utilizar la laringe, ¿se puede estimular el tracto vocal
    sólo con golpes?
08) ¿Qué utilizan las personas que perdieron la laringe para poder hablar?
09) ¿Qué se necesita para el reconocimiento del discurso oral?
10) ¿Para qué es importante el tono?

RESPUESTAS:
01) procesamiento de señales/ reconocimiento del discurso oral /
    producción del discurso oral humano
02) con un trozo de carne (grande y costoso)
03) no
04) pulmones, laringe/cuerdas vocales, cavidad articulatoria/tracto vocal
05) producir chirridos con la parte superior de un globo (de goma)
06) por el tono
07) sí
08) un vibrador en sus gargantas
09) la forma del tracto vocal
10) para reconocer las emociones (en el discurso oral)


QUESTIONS:
```

```
01) What is the topic of the lecture?
02) What does the lecturer call the human apparatus for producing speech?
03) According to the lecturer: If we had to invent a new coding strategy
    to communicate with each other, would it be identical to the one
    which we use so far?
04) What are the three components of the human apparatus for producing speech?
05) With what does the lecture compare the operation of the larynx?
06) What is the larynx responsible for?
07) Instead of using the larynx, is it possible to excite the vocal tract
    by just hitting it?
08) What are people who have lost their larynx using to be able to speak?
09) What is necessary for speech recognition?
10) What is pitch necessary for?

ANSWERS:
01) signal processing/ speech recognition/ human speech production
02) a (big expensive) piece of meat
03) no
04) lungs, larynx/vocal folds , articulatory cavity/vocal tract
05) making squeaking noises with the upper part of a (rubber) balloon
06) for the pitch
07) yes
08) a buzzer on their throat
09) the shape of the vocal tract
10) for recognizing emotions (in speech)
```

l043-2

```
PREGUNTAS:
01) ¿Qué modelo describe al tracto vocal desde un punto de vista físico?
02) ¿Cómo se denomina a la transformada utilizada para computar un espectro?
03) ¿Cuáles son las dos clases de espectros que existen?
04) ¿Qué clase de espectro es importante para el reconocimiento del discurso
    oral?
05) ¿Es posible saber qué se ha dicho con sólo mirar el espectrograma?
06) ¿Cómo se visualizan las frecuencias resonantes en un espectro gráfico?
07) ¿Cuántas frecuencias resonantes se necesitan para distinguir diferentes
    vocales?
08) ¿Cuándo se llevó a cabo esta investigación?
09) ¿Cómo se denomina el gráfico de las frecuencias formantes de las vocales?
10) De acuerdo con el disertante, ¿es posible reconocer el discurso oral con
    sólo identificar las diferentes frecuencias formantes de un espectrograma?

RESPUESTAS:
01) el modelo de tubo
02) la transformada de Fourier
03) el espectro de potencia y el espectro de fase
04) el espectro de potencia
05) sí
06) como picos
07) 2
08) en los años cincuenta y sesenta
09) el triángulo vocálico
```

10) no

QUESTIONS:
01) Which model describes the vocal tract from a physical point of view?
02) What is the transform that is used to compute a spectrum called?
03) Which two types of different spectra exists?
04) For speech recognition, which type of spectrum is interesting?
05) Is it possible by just looking at a spectrogram to tell what has been spoken?
06) How do the resonant frequencies occur in a plotted spectrum?
07) How many resonant frequencies are important to distinguish between different vowels?
08) When was this research carried out?
09) What is the plot of the formant frequencies of vowels called?
10) According to the lecturer: Can speech recognition be done by identifying the different formant frequencies in a spectrogram?

ANSWERS:
01) the tube model
02) a Fourier Transform
03) power spectrum and phase spectrum
04) the power spectrum
05) yes
06) as peaks
07) 2
08) in the fifties and sixties
09) the vowel triangle
10) no

l043-3

PREGUNTAS:
01) ¿Son 32 kilohertz la típica frecuencia de muestreo para el reconocimiento del discurso oral?
02) ¿Cuál es el objetivo del procesamiento inicial?
03) ¿Qué clase de fuentes de conocimiento utiliza el decodificador?
04) ¿Qué parte del oído humano transmite las vibraciones desde el tímpano a la cóclea?
05) Desde un punto de vista técnico, ¿qué función cumple la cóclea en el oído humano?
06) ¿Qué tipo de banco de filtro se analiza?
07) ¿Qué regla se utiliza para reformular la formulación básica del problema del reconocimiento del discurso oral?
08) ¿Se denomina modelo acústico al modelo utilizado para calcular la probabilidad de una secuencia de palabras específica?
09) ¿Qué representa la variable P?
10) ¿Cuál será el tema de la próxima conferencia?

RESPUESTAS:
01) no
02) convertir/ codificar/ comprimir la señal (en una representación compacta)
03) un modelo acústico, un modelo lingüístico/idiomático y un diccionario

```
04) el martillo
05) ejecutar la transformada de Fourier con un banco de filtro
06) el banco de filtro en la escala de Mel
07) el teorema de Bayes
08) no
09) la secuencia de palabras
10) los modelos ocultos de Markov

QUESTIONS:
01) Is the typical sampling frequency for speech recognition 32 kilohertz?
02) What is the goal of the front-end processing?
03) Which types of knowledge sources are used by the decoder?
04) Which part of the human ear passes the vibrations from the ear drum to
    the cochlea?
05) From a technical point of view, what is the cochlea doing in the human
    ear?
06) What type of filter-bank is discussed?
07) Which rule is used to reformulate the basic formulation of the speech
    recognition problem?
08) Is the model used to compute the probability of a particular word
    sequence called an acoustic model?
09) What does the variable W represent?
10) What will be the topic of the next lecture?

ANSWERS:
01) no
02) to turn/ code/ compress the signal (into a compact representation)
03) an acoustic model, a linguistic/ language model, and a dictionary
04) the hammer
05) performing a Fourier transform with a filter bank
06) Mel-scale filter bank
07) Bayes rule
08) no
09) the word sequence
10) Hidden-Markov models
```

t036+-1

```
PREGUNTAS:
01) ¿Cuál es el motivo de la charla?
02) ¿Qué significa la sigla CHIL que le da el nombre al proyecto?
03) ¿Cuántos proyectos integrados financió la Comisión Europea?
04) ¿Todos los colaboradores del proyecto CHIL forman parte de la
    Unión Europea?
05) ¿De cuántos países provienen los colaboradores del proyecto?
06) ¿Qué parte del proyecto dirige Fraunhofer?
07) ¿A cuál de las dos universidades pertenece el disertante de
    la charla?
08) ¿Cuántos años dura la primera etapa financiada del proyecto?
09) ¿Cuál es el presupuesto total para el proyecto CHIL?
10) Conforme a la charla, ¿con qué prefieren interactuar los seres
    humanos?
```

RESPUESTAS:
01) la reseña del proyecto CHIL
02) computers in the human interaction loop
03) 3
04) no
05) 9 países
06) el aspecto administrativo/ las cuestiones financieras y presupuestarias
07) Universidad de Karlsruhe y Universidad de Carnegie Mellon
08) 3 años
09) 25 millones de euros
10) otros humanos

QUESTIONS:
01) What is the occasion for the talk?
02) What does the project name CHIL stand for?
03) How many integrated projects were funded by the European Commission?
04) Are all partners, involved in the CHIL project located in the European Union?
05) From how many countries do the project partners come from?
06) Which part of the project is Fraunhofer managing?
07) With which two universities is the lecturer of the talk affiliated?
08) How many years is the length of the first funded project phase?
09) What is the total budget of the CHIL project?
10) According to the talk, what do humans prefer to interact with?

ANSWERS:
01) the review of the CHIL project
02) computers in the human interaction loop
03) 3
04) no
05) 9 countries
06) the administrative aspect / the budgetary/ financial issues.
07) University of Karlsruhe and Carnegie Mellon University
08) 3 years
09) 25 million Euro
10) other humans

t036+-2

PREGUNTAS:
01) ¿Qué se necesita para desarrollar y evaluar las tecnologías de CHIL?
02) ¿Cuáles son los dos escenarios de interacción plurimodales?
03) ¿Todos los lugares que participan de CHIL construyeron una sala CHIL?
04) ¿Dónde se encuentra ubicado el centro AIT?
05) ¿Qué idioma se seleccionó para recolectar datos?
06) ¿Dónde se registraron y transcribieron las primeras conferencias?
07) ¿Quién se ocupa de tomar notas de los datos de vídeos y discursos?
08) Enumere tres de las tecnologías mencionadas necesarias para crear servicios exitosos.
09) De acuerdo con la charla, ¿son las evaluaciones importantes para el proyecto?

10) ¿Cómo se denomina en la charla al grado de utilidad de un servicio?

RESPUESTAS:

01) datos
02) reuniones y conferencias
03) no
04) en Atenas, Grecia
05) inglés (europeo)
06) Universidad de Karlsruhe
07) ELDA y otros
08) Las tecnologías mencionadas en la charla son el seguimiento de personas, la identificación de personas, la señalización y atención, el reconocimiento del discurso oral, los sucesos acústicos, las respuesta de preguntas, el resumen como así también las subáreas del seguimiento de personas, la detección y el monitoreo del cuerpo, la detección y el monitoreo de la cabeza, el monitoreo de la mano, la localización de la fuente acústica.
09) sí
10) medida de eficacia

QUESTIONS:

01) What is needed for developing the CHIL technologies and evaluating them?
02) What are the two multi-modal interaction scenarios?
03) Have all sites participating in CHIL built a CHIL room?
04) Where is the AIT located?
05) What language was chosen for data collection?
06) At which site were the first lectures recorded and transcribed?
07) Who is performing the annotation of speech and video data?
08) List three of the mentioned technologies required for creating successful services.
09) According to the talk, are evaluations important for the project?
10) What is the measure for the usefulness of a service called in the talk?

ANSWERS:

01) data
02) meetings and lectures
03) no
04) in Athens, Greece.
05) (European) English
06) University of Karlsruhe
07) ELDA and others
08) The technologies mentioned in the talk are person tracking, person identification, pointing and attention, speech recognition, acoustic events, question answering, summarization as well as the subareas person tracking, body detection and tracking, head detection and tracking, hand tracking, acoustic source localization.
09) yes
10) measure of effectiveness

t036+-3

PREGUNTAS:
01) ¿Cuándo fue la primera evaluación de prueba?
02) ¿Dónde se llevó a cabo el taller de la primera evaluación abierta?
03) En el área de servicios, ¿qué les permiten hacer los prototipos o las demos a los socios?
04) ¿Cuántos lugares que participan del proyecto han diseñado prototipos de demostración para los servicios?
05) ¿Qué institutos trabajaron en forma conjunta para descubrir las medidas que formalizan las experiencias y la usabilidad del servicio para los usuarios?
06) ¿Cuántas tecnologías diferentes se han evaluado?
07) ¿La demostración del análisis del escenario acústico funcionó bien durante la charla?
08) En el reconocimiento del discurso oral en los noticiarios, ¿el porcentaje de reconocimiento de error fue alto o bajo?
09) Para centrarse en el monitoreo de la atención, ¿es necesario monitorear todo el cuerpo?
10) En comparación con los proyectos integrados, ¿cómo han sido los últimos proyectos europeos?

RESPUESTAS:
01) junio de 2004
02) en Atenas
03) comparar y aprender del otro
04) cuatro
05) Stanford, Universidad de Eindhoven, IRST
06) 13
07) no
08) bajo, menos del 10 por ciento
09) no
10) más cortos y de un alcance más limitado

QUESTIONS:
01) When was the first dry-run evaluation?
02) Where was the workshop of the first open evaluation held?
03) In the area of services, what do the developed demos or prototypes allow the partners to do?
04) How many project sites have built demonstration prototypes for services?
05) Which institutes worked together to come up with measures that formalize user experience and usability of services?
06) How many different technologies have been evaluated?
07) Did the demonstration of acoustic scene analysis work well during the talk?
08) For speech recognition of broadcast news, was the recognition error rate high or low?
09) For focus of attention tracking, is it necessary to track the whole body?
10) In comparison to the integrated projects, how have the past European project looked like?

ANSWERS:
01) June 2004
02) in Athens
03) compare and learn from each other
04) four
05) Stanford, University of Eindhoven, IRST
06) 13
07) no
08) low, under 10 percent
09) no
10) smaller and more limited in scope