Automatic Speech Recognition for Low-resource Languages and Accents Using Multilingual and Crosslingual Information

Zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik des Karlsruher Instituts für Technologie (KIT)

genehmigte DISSERTATION

von

Ngoc Thang Vu

aus Hanoi, Vietnam

Tag der mündlichen Prüfung:	23.1.2014
Erste Gutachterin:	Prof. Dr.-Ing. T. Schultz
Zweiter Gutachter:	Prof. E. Barnard

Acknowledgments

I would like to thank my supervisor, Prof. Tanja Schultz. She always believed in my research and supported me with many useful discussions. Her great personality and excellent research skill had a very strong effect on my scientific career. Moreover, all the travels which are one of the most beautiful experiences in my life would be not possible without her support.

Special thanks to my second supervisor Prof. Etienne Barnard who always supported my research. I am also grateful that he read my thesis and provided many useful suggestions and comments. It was very kind of him to take the long trip from South Africa to Karlsruhe to participate in the dissertation committee.

I started my PhD program at CSL in September 2009 but the first experience with speech recognition have been done in my master thesis about Vietnamese ASR. I was excited to work on speech recognition for Vietnamese, my mother tounge. Till now, I am always very grateful to all my relatives, my friends in Hanoi, and Ho Chi Minh city, Vietnam as well as in Karlsruhe, Germany to support me collecting the Vietnamese GlobalPhone data. This database and the first exiting work on automatic speech recognition motivated me to start my PhD in multilingual speech recognition.

Moreover, thanks to Roger Hsiao, I learned to build my first ASR system for French with a large amount of training data. He shared with me many experiences related to discriminative training for acoustic models.

I was in USA for the first time in 2010 when I had the chance to visit InterAct at Carnegie Mellon University and worked with Florian Metze on Bottle-Neck features. Thanks to him, I learned more about the Janus Speech recognition toolkit and Bottle-Neck features.

I was extremely fortunate to participate in the KALDI workshop in 2011 and 2013. There I got to know many new friends who are excellent researchers. The exchange with David Imseng, Stefan Kombrink, Korbinian Riedhammer, Karel Versely, Arnab Ghoshal, Martin Karafiat, Petr Motlicek, Yanmin Qian, and Sanjeev Khundapur helped me a lot. Thanks to Stefan Kombrink, I gathered the

first experience with recurrent neural network language modeling. Thanks to David Imseng, I had a better understanding of Kullback-Leibler HMM decoding. It was a great experience working with him on our first joint paper for ICASSP 2014. Furthermore, it was a great pleasure to work with Daniel Povey who had a strong effect on my research with his excellent research skills.

In 2013, I achieved the "Kontakte knüpfen" scholarship which allowed me to travel to different research groups to present my thesis and obtain feedback. Again, I had a chance to work with Daniel Povey on multilingual Deep Neural Network acoustic modeling. It was great to learn from him about deep neural networks. As a part of this tour, I also visited Nuance, ISCI and SRI International. Thanks to Sanjeev Khundapur, Paul Vozila, Korbinian Riedhammer, Andreas Stolcke, Nelson Morgan, Yik-Cheung Tam, and Dimitra Vergyri, I obtained many useful feedbacks for my dissertation.

Furthermore, I would like to thank all my friends and my colleagues at CSL for a great time. Their support is magnificent. Thanks to Tim Schlippe, Michael Wand, Matthias Janke, Dominic Telaar, Dominic Heger, Christoph Amma, Christian Herff, Felix Putze, Heike Adel, Udhyakumar Nallasamy, Dirk Gehrig and Daniel Reich for many great travel experiences and lovely activities after work. Special thanks to Tim Schlippe and Dominic Telaar for their support during difficult moments. Thanks to Franziska Kraus, Jochen Weiner, Zlatka Mihaylova, Edy Guevara Komgang Djomgang, Wojtek Breiter, Yuanfan Wang, Marten Klose and Michael Ikkert for their encouragement. Moreover, thanks to Helga Scherer for her support.

Special thanks to Heike Adel for her support and useful discussions. She was always there for me when I had a difficult time. It was also great to work together with her on language modeling for Code-Switching. I am very grateful that she read and improved all the pages of my thesis.

Finally, special thanks to my parents and my sister for their support all the time. It took more than ten years for me in Germany to obtain the diploma and the PhD in computer science. It was a very long journey and they have been always there for me.

Summary

This thesis explores methods to rapidly bootstrap automatic speech recognition systems (ASR) for languages, which lack resources for speech and language processing - called low-resource languages. We focus on finding approaches which allow using data from multiple languages to improve ASR systems for those languages on different levels, such as feature extraction, acoustic modeling and language modeling. Under application aspects, this thesis also includes research work on non-native and Code-Switching speech, which have become more common in the modern world.

The main contributions of this thesis are as follows:

Building an ASR system without transcribed audio data: In this thesis, we developed a multilingual unsupervised training framework which allows building ASR systems without transcribed audio data. Several existing ASR systems from different languages were used in combination with cross-language transfer techniques and unsupervised training to iteratively transcribe the audio data of the target language and, therefore, bootstrap ASR systems. The key contribution is the proposal of a word-based confidence score called "Multilingual A-stabil" which works well not only with well trained acoustic models but also with a poorly estimated acoustic model, such as one which is borrowed from other languages in order to bootstrap the acoustic model for an unseen language. All the experimental results showed that it is possible to build ASR systems for new languages without any transcribed data, even if the source and the target languages are not related.

Multilingual Bottle-Neck features: We explored multilingual Bottle-Neck (BN) features and their application to rapid language adaptation to new languages. Our results revealed that using a multilingual multilayer perceptron (MLP) to initialize the MLP training for new languages improved the MLP performance and, therefore, the ASR performance. Finally, visualization of the features using t-SNE leads to a better understanding of the multilingual BN features.

Improving ASR performance on non-native speech using multilingual and crosslingual information: This part presents our exploration of using multi-

lingual and crosslingual information to improve the ASR performance on non-native speech. We showed that a multilingual ASR system consistently outperforms a monolingual ASR system on non-native speech. Finally, we proposed a method called *cross-lingual accent adaptation* to improve the ASR performance on non-native speech without any adaptation data. With this approach, we achieved substantial improvements over the baseline system.

Multilingual deep neural network based acoustic modeling for rapid language adaptation: This thesis comprises an investigation of multilingual deep neural network (DNN) based acoustic modeling and its application to new languages. We investigated the effect of phone merging on multilingual DNN in the context of rapid language adaptation and the combination of multilingual DNNs with Kullback–Leibler divergence based acoustic modeling (KL-HMM). Our studies revealed that KL-HMM based decoding consistently outperformed conventional hybrid decoding, especially in low-resource scenarios. Furthermore, we found that multilingual DNN training equally benefits from simple phone set concatenation and a manually derived universal phone set based on IPA.

Multilingual language modeling for Code-Switching speech: We investigated the integration of high level features, such as part-of-speech tags and language identifiers into language models for Code-Switching speech. Our results showed that using these features in state-of-the-art language modeling techniques, such as recurrent neural network and factored language models improved the perplexity and mixed error rate on Code-Switching speech. Moreover, the interpolated language model between these two LMs gave the best performance on the SEAME database. Finally, we showed that Code-Switching is speaker dependent and, therefore, Code-Switching attitude dependent language modeling further improved the perplexity and the mixed error rate.

We believe that our findings will have an increasing impact over time not only for research but also for industry. The results can be used to save costs and developmental time for the building of a speech recognizer for a new language. In addition, the contribution of this thesis on non-native and Code-Switching speech will become more important due to the rapidly growing globalization.

Zusammenfassung

In dieser Arbeit erforschen wir verschiedene Methoden, um automatische Spracherkennungssysteme (ASR) für neue Sprachen mit wenigen Ressourcen zu entwickeln. Insbesondere konzentrieren wir uns auf Ansätze, Daten aus mehreren Sprachen zu verwenden, um verschiedene Komponenten der ASR solcher Sprachen wie Merkmalsextraktion, akustische Modellierung und Sprachmodellierung zu verbessern. In Bezug auf Anwendungen beinhaltet diese Dissertation auch Forschungen über akzentbehaftete und Code-Switching Sprache, die in der modernen Welt immer häufiger vorkommen.

Die wichtigsten Beiträge dieser Arbeit sind die folgenden:

Aufbau eines ASR-Systems ohne transkribierte Sprachdaten: In dieser Arbeit wird ein multilinguales, unüberwachtes Trainingsframework entwickelt, das den Aufbau eines ASR-Systems ohne transkribierte Daten ermöglicht. Idee ist es, Spracherkenner anderer Sprachen in der Kombination mit unüberwachtem Training zu verwenden. Dadurch werden die Zeit und Kosten für das Transkribieren der Sprachdaten minimiert. Ein wesentlicher Beitrag ist die Entwicklung eines wortbasierten Konfidenzmaßes namens "multilingual A-stabil", das nicht nur mit robusten akustischen Modellen, sondern auch mit einem schwachen akustischen Modell funktioniert. Alle experimentellen Ergebnisse zeigen, dass wir ein ASR-System für neue Sprachen ohne transkribierte Daten bauen können, selbst wenn die Quell- und Zielsprachen nicht verwandt sind.

Multilinguale Bottle-Neck Sprachmerkmale: Die Integration von neuronalen Netzen in die Vorverarbeitung des Spracherkenners in Form von Bottle-Neck Merkmale ist Stand der aktuellen Forschung. In dieser Arbeit werden multilinguale neuronale Netze und ihre Anwendbarkeit für neue Sprachen untersucht. Wir stellen einen innovativen Ansatz vor, der zur Initialisierung bereits trainierte multilinguale neuronale Netze verwendet. Eine Visualisierung der Merkmale mittels t-SNE erlaubt es, ein besseres Verständnis für multilinguale Bottle-Neck Sprachmerkmale zu entwickeln.

Verbesserung der ASR Leistung auf akzentbehafteter Sprache mit Hilfe von multilingualen und crosslingualen Informationen: Diese Arbeit erforscht die Verwendung von multilingualen und crosslingualen Informationen zur Verbesserung der ASR Leistung auf akzentbehafteter Sprache. Wir zeigen, dass ein multilinguales ASR-System auf akzentbehafteter Sprache besser funktioniert als ein monolinguales ASR-System. Außerdem haben wir eine neue Methode, *crosslingual accent adaptation*, entwickelt, die die ASR Leistung ohne Adaptionsdaten auf akzentbehafteter Sprache verbessert. Mit diesem Ansatz konnten wir signifikante Verbesserungen gegenüber dem Referenzsystem erreichen.

Akustische Modellierung basierend auf multilingualen Deep Neural Networks: Diese Arbeit umfasst die Untersuchung multilingualer Deep Neural Network (DNN) für akustische Modellierung und ihre Anwendung auf neue Sprachen. Wir untersuchen den Effekt der Verschmelzung des Phonesets beim Training eines DNNs und der Kombination von multilingualen DNNs mit Kullback-Leibler Divergenz Hidden Markov Model (KL-HMM) beim Dekodieren auf die ASR Leistung bei neuen Sprachen. Unsere Untersuchungen zeigen, dass KL-HMM basierte Dekodierung die ASR Leistung verbessert, insbesondere wenn Trainingsdaten für die neue Sprache nur eingeschränkt vorhanden sind. Weiterhin haben wir festgestellt, dass die Verschmelzung des Phonesets auf IPA-Basis keinen Effekt auf das multilinguale DNN Training hat.

Multilinguale Sprachmodellierung für Code-Switching Sprache: Wir untersuchen die Integration von linguistischen Merkmalen wie Wortarten und Sprachidentifikatoren in Sprachmodelle für Code-Switching. Unsere Ergebnisse zeigen, dass die Verwendung dieser Merkmale in verschiedenen Sprachmodellierungstechniken, wie z.B. rekurrente neuronale Netze oder faktorisierte Sprachmodelle, die Perplexität des Sprachmodells und auch die Fehlerrate des Spracherkenners auf Code-Switching verbessert. Außerdem liefert die Kombination dieser beiden Techniken die beste Leistung auf unserem Testset. Schließlich zeigen wir, dass Code-Switching-Verhaltens sprecherabhängig ist. Daher liefert Code-Switching verhaltensabhängige Sprachmodellierung weitere Verbesserungen auf dem Code-Switching Datenkorpus.

Die Bedeutung dieser Dissertation wird in Zukunft nicht nur in der Forschung sondern auch in der Praxis steigen. Zum einen können die Ergebnisse genutzt werden, um Kosten und Entwicklungszeit für den Bau eines Spracherkenners für eine neue Sprache zu sparen. Zum anderen gewinnen die Arbeiten mit akzentbehafteten Sprachen und Code-Switching mehr Bedeutung aufgrund der schnell zunehmenden Globalisierung.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Human-machine communication is one of the most important research fields in the last decade. Speech processing is an important subarea, since speech is the most natural way of human communication. In addition, due to globalization the need of communication across language barriers increases. Therefore, research on multilingual speech processing becomes important and earns a lot of attention in the research community and the industry. This thesis deals with the topic of multilingual speech recognition. The main research ideas are presented in this chapter.

1.1 Aspects of multilingual ASR

Automatic speech recognition (ASR) is called multilingual if at least one of the components, such as feature extraction, acoustic model, pronunciation dictionary or language model is created by using data of multiple languages - *multilingual data*. Since multilingual data are used, the linguistic knowledge can be shared and transfered between languages. Therefore, multilingual ASR is suitable to applications, in which

- The target languages lack resources (low-resource languages).

- The acoustic or linguistic characteristics of two languages impact each other (non-native speech).
- Multiple languages appear in a conversion or an utterance, such as Code-Switching speech.

The next following paragraphs explain the challenges of low-resource languages, non-native speech and Code-Switching in more details. Moreover, the role of these terms in this thesis is characterized.

Based on the availability of resources, languages can be categorized in well-resource languages and low-resource languages. While more than 6,900 languages exist all over the world, the number of well-resourced languages is quite limited. Most speech processing systems can only handle very few languages. Google Voice Search, for example, includes 29 languages and accents (2012). Further core systems today are Siri ASR application with 8 languages (2012) and Dragon with 40 languages (2013). The gap from those few languages to 6,900 languages in the world has its most important reason in different availabilities of resources. A large amount of languages are low-resource. The term low-resource refers to languages with one or more of the following aspects: lack of a unique writing system or stable orthography, lack of linguistic expertise, lack of electronic resources for speech and language processing. If the goal is to rapidly bootstrap ASR systems for new languages, the first immediate step is to concentrate on low-resource languages which lack of resources for speech and language processing, such as transcribed speech data. In this thesis, low-resources languages with small amounts of transcribed audio data or no transcribed audio data at all are addressed.

Accented speech is a very important application of multilingual ASR. More specifically, an accent is a manner of pronunciation peculiar to a particular individual, location, or nation [Dic05]. An accent may identify the locality in which speakers reside (a regional or geographical accent), the socio-economic status of its speakers, their ethnicity, their caste or social class (a social accent), or influence from their first language (a foreign accent) [LG97]. This thesis focuses only on the challenges of foreign accents, which are known as "non-native speech". For example, a Chinese speaking English will sound different compared to an American or a Britain speaking English. In this case, the Chinese speaker is a non-native speaker and English is not the mother tongue. The mother tongue of the speaker could be Mandarin or Cantonese, which is referred to as *L1*. English is another language which the speaker can speak. It is called *L2*. For many years, non-native speech has been a big challenge for state-of-the-art ASR systems. Two of the main challenges of ASR for non-native speech are high phonetic variations among speakers depending on the their mother tongue and their proficiency level, and lack of resources, such as transcribed audio data.

Another important application of multilingual ASR is the recognition of Code-Switching speech in which multiple languages can appear. CodeSwitching speech is a common phenomenon in multilingual communities. Its main characteristic is that speakers change languages during a conversation or even within a sentence. The main challenges of ASR for Code-Switching is the lack of bilingual training data. Moreover since the speakers use multiple languages in a conversation, the pronunciation may be changed due to co-articulation effects. Due to the characteristics of Code-Switching speech, multilingual ASR is one of the most suitable solutions.

1.2 History of multilingual ASR

Multilingual speech recognition has a long research history in the speech recognition community starting in the late nineties. There are many studies which followed this research direction and demonstrated successful results. However for a long time, multilingual speech recognition seemed to be interesting only for the academic world. This situation has changed dramatically as will be explained later in part 1.3.

The following paragraphs provide an overview of the beginnings of using multilingual and crosslingual information in speech recognition systems.

In the preprocessing step, cepstral features were widely used as speech features. Since they are assumed to be language independent, there was no reason to conduct research on using multilingual data for feature extraction. However in 2002, new features for speech recognition were introduced by H. Hermansky which are called Tandem features [HDS00]. They use the output of a neural network which has many hidden layers called multilayer peceptron (MLP) for the speech recognition task. The neural network uses the cepstral features as input and is trained on transcribed audio data. After that, researchers investigated the use of crosslingual and multilingual data to train the neural network and, therefore, improve the Tandem features for the speech recognition task. Several studies showed that features extracted from an MLP which was trained with one language or multiple languages can be applied to further languages [CMDL$^+$07, TFGK08, PSN11].

In the late nineties, researchers started to systematically investigate the usefulness of language independent acoustic models to bootstrap systems to unseen languages. Studies especially considered the impact of language families ([CC97]), the impact of the amount of languages used to create acoustic models ([GG97], [SW98a]), the impact of the amount of training data ([WKAM94, Köh98, SW98b]) and possible ways to share acoustic models across languages

([SW98b, Köh98]). One of the early findings was that multilingual acoustic models outperform monolingual ones for the purpose of rapid language adaptation ([SW01b]).

In the context of multilingual language modeling, there are only few previous studies. Several research in the late nineties concentrated on building language models to handle switches between languages in a sentence [CDG+97, AHG+98, WRN+98] or between sentences [WBNS97]. In later research since 2002, the investigation of the transfer of information which appears in one language to other languages using dictionary-based translation models was presented in [KK02]. Furthermore, methods were developed which allowed the combination of several monolingual models into one multilingual language model [FSS+03].

1.3 Current developments

Compared to the late nineties, the situation has dramatically changed. The economic, technological, sociocultural, and political sectors have been changed during the last decade by a process commonly referred to as globalization. Moreover, the use of Internet increases rapidly all over the world. Due to the these facts, the availability of multimedia data and the need of multilingual applications have changed. Applications with speech technology are used not only in industrial countries, such as the United States, Germany or Japan but also in developing countries, such as Thailand, Vietnam or South Africa. Naturally, people prefer to use their mother tongue to communicate with each other or with machines. Therefore, there is an urgent need of supporting many languages. Furthermore, with the strong growth of the Internet, divers media provide a great amount of easily and inexpensively accessible audio data for various languages. However, there are no restrictions in topic or vocabulary for those data, and one has to deal with different dialects or even different languages. Moreover, the most crucial problem is the possible lack of transcriptions. To overcome these limitations, automatic methods for training a speech recognition system which does not require transcribed audio data are necessary. Moreover, methods are required which allow using those data more efficiently to train multilingual models which can be used to bootstrap and improve an ASR system for a new language or accent. Finally, as a part of globalization, the exchange of economy, technology and migration occurs more often and easier than in the past, e.g. multilingual communication becomes more popular over the world. There are more and more non-native speakers who use speech technology for their multilingual communication. Hence, the need

of developing an ASR system which can handle non-native speech is more important than in the past. Moreover, bilingualism is more common in different countries, such as Singapore, Malaysia, South Africa, USA, or India. This involves that people switch language while they communicate (Code-Switching). Indeed, Code-Switching is a challenging task for state-of-the-art speech technology since there has not been a lot of research in this direction yet.

To sum up, due to the rapid changes of the initial situation in the last fifteen years, multilingual speech recognition becomes more important and earns attention not only in the academic but also in the industrial world. The building of an ASR system for a new language with minimal human effort is a very important research topic. The success of approaches for this will save a lot of time and costs in the development of ASR systems for many languages. As a result, it will be possible to increase the usage of speech technology applications around the world. Moreover, an ASR system which can be used to handle special multilingual challenges, such as non-native or Code-Switching speech is necessary.

1.4 Main contributions

1.4.1 Objectives

The most important goal of this thesis is the exploration of methods to use multilingual and crosslingual information to rapidly bootstrap and improve an ASR system for low-resource languages. First, we address the case that no transcribed audio data is available. We aim at developing a training framework which allows using ASR systems from several resource-rich languages and available data resources of the target language, such as language model, pronunciation dictionary and untranscribed audio data. With this framework, it is possible to automatically build an ASR system for the target language with minimal human effort. Afterwards, we focus on finding approaches which allow sharing data across multiple languages to improve the ASR system in different levels, such as feature extraction, acoustic modeling and language modeling.

Furthermore, under application aspects, this thesis includes research work on non-native and Code-Switching speech, which have become more common in the modern world. First, we aim at exploring systematically how to improve ASR performance on non-native speech with and without adaptation data using multilingual and crosslingual information. For the application of

Code-Switching speech, we concentrate on the investigation of language modeling. Our goal is to integrate linguistic knowledge into state-of-the-art language modeling techniques to build a multilingual language model which predicts not only the next word but also the switches between languages.

1.4.2 Contribution

The main contributions of the thesis are as follows:

1. Development of a multilingual unsupervised training framework which allows training an ASR system for a new language without any transcribed audio data: Several ASR systems from different languages (source languages) are used to bootstrap an ASR system for a new language (target language) for which the pronunciation dictionary, the language model and untranscribed audio data are given. We propose a new method to compute a word-based confidence score called "multilingual A-stabil" which works well not only with well trained but also with poorly estimated acoustic models. We present our multilingual unsupervised training framework which uses all the available resources to train an ASR for new languages automatically. We demonstrate that the framework generalizes well and, thus allows building ASR systems for many languages even if the source and the target languages are not related. To our knowledge, this has never been shown in the literature before.

2. Study of a method to extract Bottle-Neck features for low-resource languages using a multilingual multilayer perceptron (MLP): The key idea is to use a multilingual MLP which can be trained with a large amount of training data from different languages as an initial model to bootstrap an MLP for a new language. For both, large and a very small amounts of data, we demonstrate that the performance of the new MLP and, therefore, the final ASR performance are significantly improved. Moreover, our research reveals that the number of languages, and the amount of data as well as the similarity of the source and target language have a strong impact on the final ASR performance. Last but not least, we showed that visualization of the features using t-Distributed Stochastic Neighbor Embedding [VdMH08] leads to a better understanding of the multilingual BN features.

3. Investigation of the use of multilingual and crosslingual information to improve ASR performance on non-native speech: First, if the adaptation data is available, our experimental results show that bilingual L1-L2 acoustic models can improve ASR performance on non-native speech.

If information of L1 or L1 data is not available, multilingual ASR outperforms monolingual ASR on non-native speech. Second, for the case that no adaptation data for the target accent is available, we propose an innovative method called *crosslingual accent adaptation* which allows sharing adaptation data across L2 languages with the same non-native accent. This proposed approach provides significant improvements over the baseline system on the non-native test data without any adaptation data. To our knowledge, this has never been shown before in literature.

4. Multilingual deep neural network based acoustic modeling for rapid language adaptation: We investigate the effect of IPA based phone merging on the multilingual DNN and its application to new languages. Moreover, multilingual DNNs in combination with Kullback-Leibler decoding in the context of rapid language adaptation for low-resource languages are explored. On different languages, we find that Kullback–Leibler divergence based hidden Markov models in combination with crosslingual model transfer yields the best performance. Furthermore, our experiments suggest that it is not necessary to manually derive IPA based universal phonesets for multilingual DNN training.

5. Exploration of multilingual language modeling in context of Code-Switching (CS) speech: We propose a method to train a multilingual language model which can be used for Code-Switching. Different features, such as Part-Of-Speech tags (POS) and language identification (LID) are integrated into Recurrent Neural Network language models and Factored language models to predict not only the next word but also the switches between languages. Furthermore, our analyses of Code-Switching points show that the Code-Switching phenomenon is speaker dependent and there are several groups of speakers which share the same "Code-Switching attitude".

1.5 Structure of the thesis

This thesis is organized as follows:

Chapter 2 (*Background*) provides a brief introduction into the field of automatic speech recognition. Cepstral features and multilayer perceptron features are presented. Basic techniques, such as HMM/GMM and advanced techniques like Deep Neural Network are briefly described. State-of-the-art language modeling techniques, such as N-gram language models, factored language models and recurrent neural network language models are presented and compared. Furthermore, lattices and N-best lists are explained. We also

describe the unsupervised acoustic model training and adaptation approaches which are relevant to this thesis.

Chapter 3 (*Data, Tools and Baseline (ASR) Systems for Multiple Languages*) describes the resources including the databases which are used for the experiments and the baseline monolingual ASR systems. The database part includes the descriptions of the GlobalPhone data, the non-native speech corpus, the VOV database and the SEAME corpus. Finally, we present our monolingual ASR systems for many languages which were built with GlobalPhone data. Those ASR systems serve as baseline in many experiments in this thesis.

Chapter 4 (*Cross-language Bootstrapping Based on Completely Unsupervised Training*) describes our multilingual unsupervised training framework (MUT) which allows training an ASR system for a new language without any transcribed data. First, we revisit the cross-language transfer techniques and investigate the correlation between the ASR performance and the similarity between source and target language. Second, we present a new method to compute confidence scores called "multilingual A-stabil" which works quite well not only with well trained acoustic models but also with poorly estimated acoustic models. In the experiments, we apply our framework MUT to build ASR systems for different scenarios with increasing levels of difficulty.

Chapter 5 (*Multilingual Bottle-Neck Features and Their Application To Low-resource Languages*) presents our investigation on using multilingual data to improve multilayer perceptron features for new languages. The study starts with our proposal of using multilingual MLPs to initialize the monolingual MLP training which allows training an MLP with a very small amount of training data. Afterwards, we explore the correlation between the similarity of source and target languages and the final ASR performance. Finally, this chapter ends with a visualization of the output of the bottle-neck hidden layer to provide a better understanding of the behavior of those features in the context of multilingual and crosslingual characteristics.

Chapter 6 (*A Study on Using Multilingual and Crosslingual Information To Improve Non-Native ASR*) describes the investigation of automatic speech recognition (ASR) on non-native speech. We explore the effect of multilingual acoustic modeling on non-native speech in different ways. First the bilingual acoustic models trained with L1 and L2 training data are evaluated on non-native speech. For the case that L1 is unknown or L1 data is not available, a multilingual ASR system trained without L1 speech data is examined. Finally, we propose a method called *crosslingual accent adaptation*, which allows using English with Chinese accent to improve the German ASR on a German with Chinese accent.

Chapter 7 (*Multilingual Deep Neural Network based Acoustic Modeling For Rapid Language Adaptation*) investigates the effect of IPA based phone merging on multilingual DNNs in the context of rapid language adaptation. We also explore the multilingual DNNs in combination with KL-HMM decoding to improve ASR accuracy. Furthermore, the influence of different pre-training methods on crosslingual DNN based acoustic modeling is studied.

Chapter 8 (*Multilingual Language Model for Code-Switching Speech*) describes the investigation of language modeling for Code-Switching on the SEAME corpus. We present different analyses of textual features which might have potential to predict Code-Switching. A recurrent neural network language model (RNNLM) and a factored language model (FLM) are used to improve the LM performance on Code-Switching speech. Additionally, we present an analysis which shows that RNNLM and FLM provide complementary information. Hence, the linear interpolation of RNNLM and FLM provides the best performance on the SEAME corpus. Finally, the investigation on *Code-Switching attitudes* is presented.

CHAPTER 2

Background

This section gives an overview of two fundamental backgrounds for the thesis. First, the languages of the world are described. In particular, the following two questions are discussed: How many languages are spoken in the world? How can the similarity between languages be estimated? Second, state-of-the-art techniques of automatic speech recognition including preprocessing, acoustic modeling, language modeling and some advanced techniques, such as unsupervised training and acoustic model adaptation are introduced.

2.1 Languages

2.1.1 Languages of the world

The question how many languages are spoken in the world is interesting, albeit difficult. One reason why the question is not easy to answer is that the number of languages changes over time. Another reason is that the opinion which dialect is considered as language might change. For example in 1996, the edition of Ethnologue listed 6,703 languages distributed over the five continents. The 2009 edition listed 6,909 living languages. However, those 206 more languages might not have been created over the years. Rather, the decision of the linguistic communities about how to distinguish languages might have changed.

In terms of number of speakers, we observe a range from 867 million native speakers (of Mandarin Chinese) down to 1 or 2 speakers (of Coos in Southern Oregon). Table 2.1 lists the top 20 languages by the number of speakers according to [Gor].

Table 2.1: *Top 20 languages sorted by the number of speakers [Gor]*

Rank	Language	Speakers (in millions)	Rank	Language	Speakers (in millions)
1	Mandarin	867.2	11	Wu	77.2
2	Spanish	322.3	12	Javanese	75.5
3	English	309.4	13	Telugu	69.7
4	Arabic	206.0	14	Marathi	68.0
5	Hindi	180.8	15	Vietnamese	67.4
6	Portuguese	177.5	16	Korean	67.0
7	Bengali	171.1	17	Tamil	66.0
8	Russian	145.0	18	French	64.8
9	Japanese	122.4	19	Italian	61.5
10	German	95.4	20	Urdu	60.5

Many of the about 6,000 languages mentioned in Ethnologue are endangered or nearly extinct. They have less than 10,000 speakers which makes them especially vulnerable. For about half of the world's languages, new generations of children are not being raised to speak them anymore. Hence in the future, the number of languages in the world may be reduced very much. As a result, efforts should be taken to preserve languages [UNE13].

2.1.2 Linguistic description and classification

This section summarizes relevant information about the linguistic description and classification based on [SK06]. Languages can be classified based on historical relatedness (language family) and linguistic characteristics (typology). These two criteria are not always correlated. English and German, for example, are North Germanic languages, but have a very different word order. English almost always uses SVO (Subject-verb-object) order while German puts the V (verb) at the end of relative clauses.

From the point of view of speech technology, studies about the relativeness between languages can be very useful. For languages which share the same characteristics, the same speech and language processing techniques can be applied to achieve better performance.

Language families

The establishment of family trees charting the genetic relatedness of languages has been a concern of historical linguistics for a long time, and there has been much debate about the categorization of particular languages within this scheme. More details of the genetic classification of languages can be found in [Kat02]. According to [Kat02], there are 21 major language families. Figure 2.1 shows their distribution over the world. The five largest and most widely known language families are Indo-European, Afro-Asiatic, Niger-Congo, Sino-Tibetan and Austronesian. In addition to these, there are many small groups, such as Dravidian, Australian, and American Indian languages, as well as many "independent" languages, such as Basque (language spoken in northern Spain) or Aimu (language spoken on Hokkaido island of Japan).

The *Indo-European* family is the world's largest family in terms of number of speakers and contains almost all the languages spoken in Europe plus many languages in India and the Middle East. Figure 2.2 illustrates the Indo-European language tree which has eight main branches, namely Germanic, Italic, Romance, Celtic, Hellenic, Slavic, Baltic, and Indo-Iranian. In this thesis, several languages from Germanic, Romance and Slavic language families were used.

The second largest language family is the *Sino-Tibetan* family which contains more than 400 languages spoken in East Asia, Southeast Asia and parts of South Asia, including the Chinese and Tibeto-Burman languages. In this thesis, three languages of *Sino-Tibetan*, namely Mandarin, Thai and Vietnamese are used in our experiments. Note that Asian languages are distributed over different language families. For example, Japanese and Korean do not belong to the *Sino-Tibetan* but to the *Altaic* language family.

Language topology

Using language *typology* is another way to classify languages into different categories. This classification is based on structural characteristics. This subsection concentrates only on those linguistic characteristics which are relevant to the speech technology, such as sound structure, word formation and sentence structure.

Phonetics, phonology and prosody describe the sound structure of a language. While the goal of *phonetics* is the analysis of sound acoustics, sound production and perception, *phonology* studies the functional, contrastive role of sounds in an entire system. In contrast, *prosody* studies concentrate on pitch, stress, intonation, and phrasing that span several sound segments. Sounds as specific

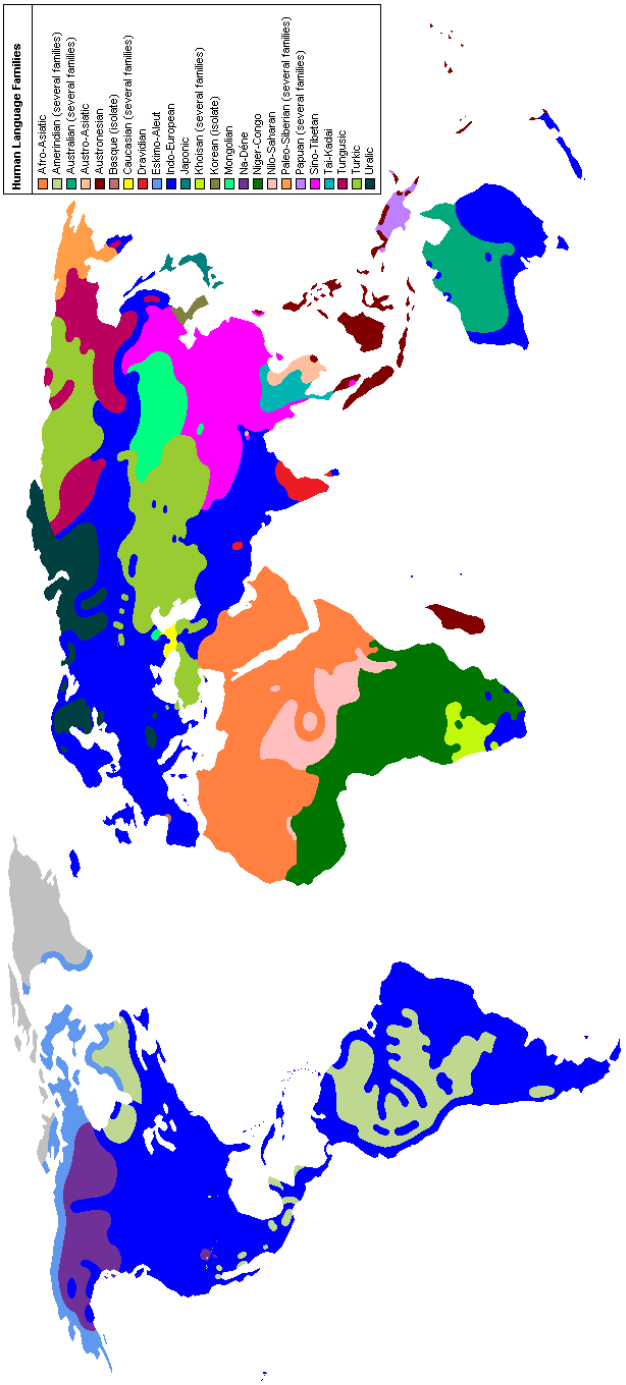
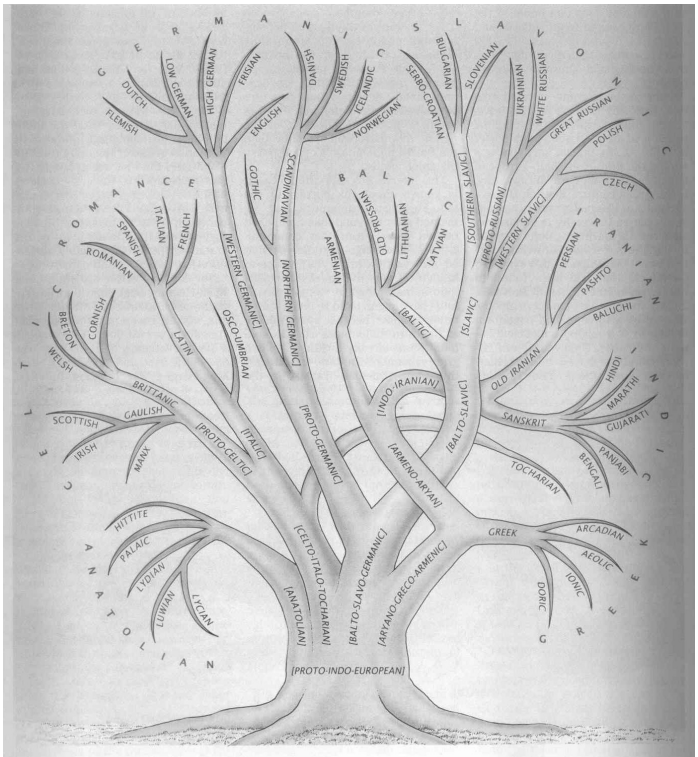


Figure 2.1: The distribution of language families over the world [Wik13]



FAMILY TREE of the Indo-European languages can be traced back to a protolanguage that flourished more than 6,000 years ago. The protolanguage split into dialects, which evolved into distinct languages; these then fissioned into generations of daughter languages. Tocharian, a dead language of Asia, has ties to Celtic, an ancient European tongue. Similarities between the Balto-Slavic and Indo-Iranian families indicate that they influenced each other before their speakers moved north and south, respectively. Dead languages are shown in italics; languages that left no literary remains are enclosed in brackets.

Figure 2.2: *Indo-European* language tree [GI90]

acoustic events are referred to as *phones* while sounds as smallest units that distinguish a minimal word pair are called *phonemes*.

The description of phones in terms of articulation is usually based on the categories of the International Phonetic Alphabet (IPA) [Ass99]. IPA is an internationally used notational system for transcribing speech sounds. It was first proposed in 1988. One of the most commonly used versions is shown in figure 2.3. Under this scheme, phones are classified based on their phonetic features. For *Consonants*, they use 1) their place of articulation (bilabial, labiodental, dental, alveolar, postalveolar, retroflex, palatal, velar, uvular, pharyngeal, glottal), 2) their manner of articulation (plosive, nasal, trill, tap/lap, fricative, lateral fricative, approximant, katerak approximant) and 3) their phonation type (voiced

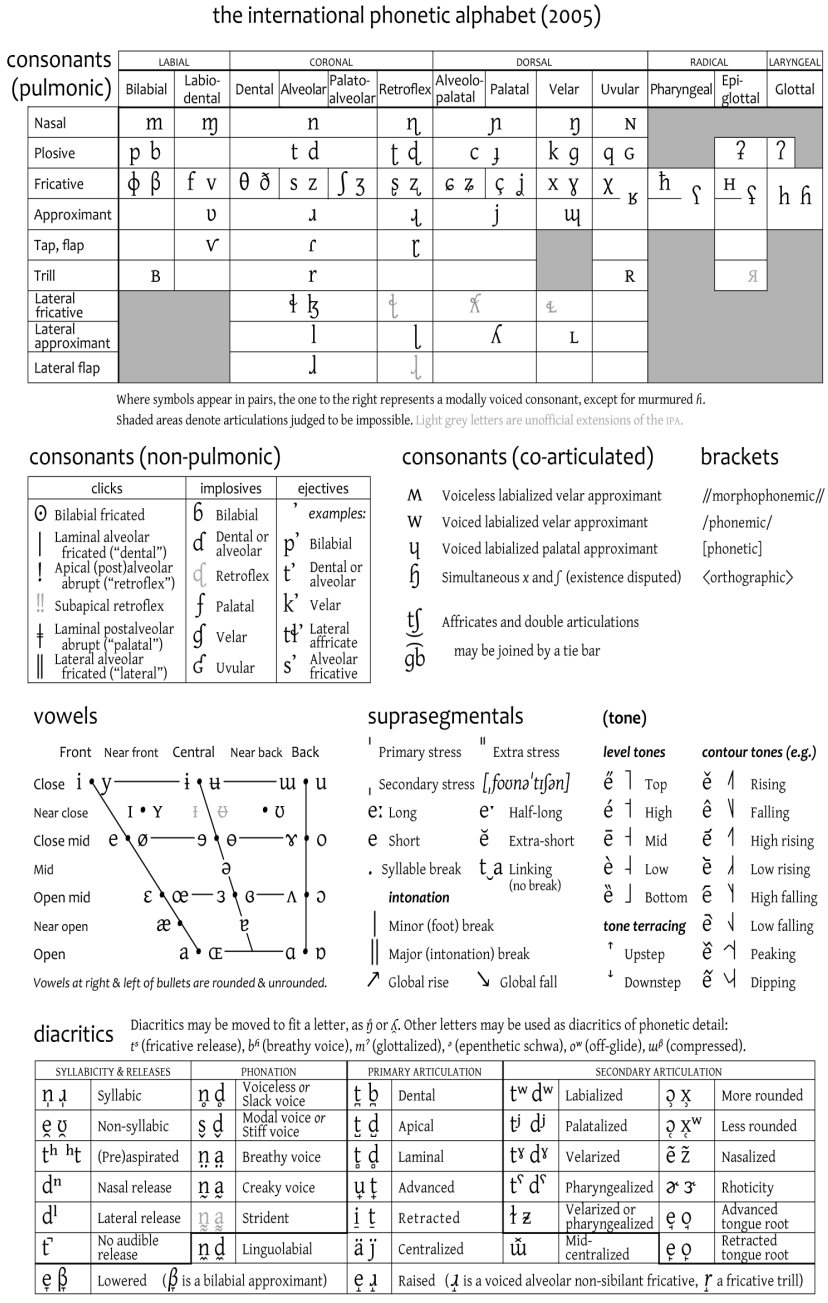


Figure 2.3: The International Phonetic Alphabet (IPA) [Ass99]

or voiceless). *Vowels* are classified based on tongue height, tongue advancement, lip rounding, and nasality. Moreover, the voice quality and the length of vowels are also important features. Each language has a phoneme inventory which indicates the complexity of the language. In addition to the phoneme inventory, the pattern of phoneme combinations is also a feature to classify the language.
At the prosody level, *pitch*, *duration* and *rhythm* are important phenomena. *Pitch* denotes the fundamental frequency of sounds. It can be used in two major ways: in *tonal languages* and *intonation languages*. In the case of *tonal languages*, the pitch contours give different meaning to the words, e.g. in Mandarin or Vietnamese. By contrast, *intonation languages* use pitch contours to indicate phrase and sentence boundaries, and for contrastive emphasis.

Morphology describes the process of the word formation in a language in which the smallest meaningful parts of the language (*morphemes*) are combined in order to form larger words. Languages can be classified based on their word formation mechanisms. The class of *isolating languages* simply forms sequences of invariable free morphemes. Such languages are often said to "have no morphology". Vietnamese is one of those languages. There is no clear segmentation between words or word boundaries. White spaces occur directly after each morpheme and each morpheme could be accepted as an individual word. *Agglutinative languages* combine several morphemes per word and each morpheme can be identified by a linear segmentation of the word into its components. Examples for those languages are Turkish and Tamil. Another class is *fusional languages* which also uses several morphemes per word. However, compared to the *agglutinative languages*, the combination of morphemes within a word may lead to a new word form. Most languages belong to more than one of the three categories described above.

Word order refers to the properties of a phrase and the sentence structure (*syntax*). It is most often categorized by the relative ordering of subject (S), verb (V), and object (O). The six resulting possible word orders - SOV, SVO, VSO, VOS, OVS, and OSV - cover all the languages in the world. However, the first two types have a much higher frequency than the others. Most languages do not have only one of these types but also allow several different word orders. Some languages like German or Russian have a "free" word order since all the word orders are possible. Moreover, it is hard to say for those languages which order is more frequent than another.

For speech processing applications, the morphological complexity of a language and the number of possible word orders are important to state the difficulty of the language modeling task.

2.2 Automatic speech recognition

The fundamental problem of speech recognition is to find the most likely word sequence given a speech recording. The following equation which is based on Bayes' rule summarizes the mathematical model commonly used for large vocabulary continuous speech recognition (LVCSR):

$$\hat{W} = argmax_w P(W|X) = argmax_w \frac{P(W)P(X|W)}{P(X)} \tag{2.1}$$

As a result of the digital signal processing, the acoustic signal is represented as a sequence of acoustic vectors X that capture the most important information of the speech signal for the classification task. The goal is to estimate the most likely word sequence $\hat{W} = W_1, W_2, ..., W_n$ depending on the prior probability $P(W)$ provided by a language model and the conditional probability $P(X|W)$ given by an acoustic model. Since the language model works on word level and the acoustic model on acoustic units like phones, a pronunciation dictionary is required to bridge the gap between words and phones. The pronunciation dictionary used for LVCSR systems is a mapping between words and their pronunciations. For the computation of the most probable word sequence, the denominator $P(X)$ is not considered since it is irrelevant for maximizing the function. Finally, to find the word sequence with the highest probability ($argmax_w$), a search strategy has to be applied. The following subsections describe the preprocessing, acoustic modeling, and language modeling in more detail.

2.2.1 Signal preprocessing

Cepstral features

Goal of the signal preprocessing step is to extract features from the speech signal which provide a compact representation of speech. They are calculated by dividing the speech signal into smaller blocks (typically between 10 and 30 ms). It is a common practice to let the blocks overlap and extend their duration (e.g. 16ms, 25ms). There are different ways of extracting speech signal features. In LVCSR, commonly used features are the *Mel-Frequency Cepstral Coefficients* (MFCCs) [DM80]. MFCCs are the representation of the short-term power spectrum of a sound wave, transferred on the Mel scale by using overlapping triangular windows. Another way to extract information about the sound spectrum is perceptual Linear Prediction (PLP) coefficients [Her90]. PLP computes linear prediction coefficients from a perceptually weighted non-linearly compressed power spectrum and, then, transforms the linear prediction coefficients

to cepstral coefficients. In addition to spectral coefficients, first order (delta) and second order (delta-delta) regression coefficients are often used to capture the temporal changes in the spectra.

Multi Layer Perceptron features

In the last years, the use of neural networks to improve ASR performance earned a lot of attention in the speech community. One application of them is using multilayer perceptrons (MLP) for feature extraction. Instead of cepstral features, the values of the output layer (Tandem features [HDS00]) or the values of the hidden layer (Bottle-Neck features [GKKC07]) are used in the preprocessing step. In many setups and experimental results, MLP features proved to be of high discriminative power and very robust against speaker and environmental variations. Figure 2.4 shows the layout of an MLP architecture which has been adopted from [MHJ+10]. As input for the MLP network, eleven stacks of adjacent MFCC feature vectors of 13 dimensions each can be used. To train the MLP, phones, subphones or context dependent subphones (details in 2.2.2) can be applied as target classes. The network has several hidden layers. One of them has a significant smaller number of neurons compared to the rest. This layer is called Bottle-Neck (BN) layer. Only the output of the BN layer is used for the speech recognition task. This also means that only the first hidden layers up to the BN layer need to be stored on the disk to extract the final speech features. Since the MLP is trained to discriminate among speech units, the output of the BN layer is expected to condensate the most important information of the MFCC features for the classification task.

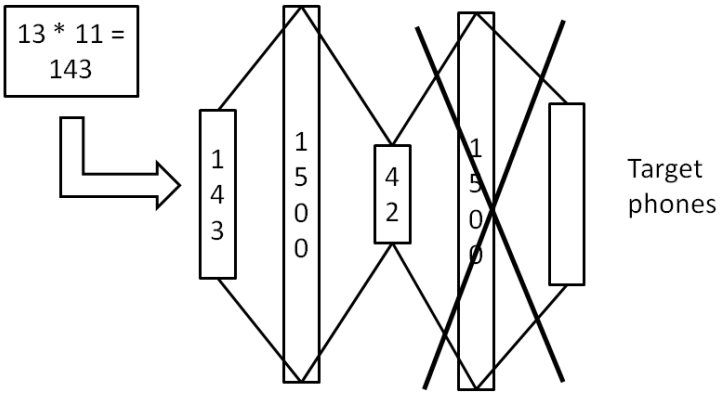


Figure 2.4: Bottle-Neck feature

Feature dimension reduction

To increase the context information at the feature level, the cepstral or bottleneck features are usually stacked with a certain number of left and right neighboring frames. However, stacking significantly increases the feature dimension which can lead to data sparsity problems and increases the confusion ability among classes. Therefore, different feature dimension reduction techniques can be applied to extract the final features. The most widely used technique in speech recognition is linear discriminant analysis (LDA) [Fuk90]. It aims at finding a linear combination of features which separates two or more classes. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before classification. First, the features are assigned to their corresponding classes. Afterwards, the LDA matrix is estimated to minimize the variance within a class and maximize the variance between classes, which is also known as *Fisher criterion*. Hence, it results in a projection which separates the classes as much as possible while increasing their compactness at the same time. Therefore, the final features are discriminative and suitable for a classification task.

2.2.2 Acoustic modeling

Hidden Markov Model (HMM)

In LVCSR, the acoustic is modeled by using smaller units than words, like phones, subphones or context dependent subphones (senones). *Hidden Markov Models* (HMM) [Rab89] are currently the most widely used representation of those units. An HMM λ is a 5-tuple consisting of the following elements:

- Set of states $S : S_1, S_2, .., S_N$. In any discrete moment, the system is in one of these states. In comparison to a Markov Model, the current HMM state is unknown or "hidden". Observing the system leads to an indirect conclusion in which particular state the system may be at a certain time.
- A discrete alphabet $V : v_1, v_2, .., v_M$ of possible emissions.
- State transition probability distribution matrix A, where a_{ij} is the probability of moving from state S_i to state S_j in the next step given a current state S_i.
- A matrix B of the emission probability distribution $(b_j(k))$ where $b_j(k)$ denotes the probability of emitting symbol v_k in state S_j.
- The probability distribution π that assigns a probability to each state S_i to be the initial state.

In the first-order HMM, there are two assumptions. The first assumption is the Markov assumption:

$$P(s_t|s_1^{t-1}) = P(s_t|s_{t-1}) \tag{2.2}$$

where s_1^{t-1} represents the state sequence $s_1, s_2, ..s_{t-1}$. Hence, this assumption states that the probability for the next state only depends on the previous state and not on the states before that.

The second is the output-independence assumption:

$$P(X_t|X_1^{t-1}, s_1^t) = P(X_t|s_t) \tag{2.3}$$

where X_1^{t-1} represents the output sequence $X_1, X_2, .., X_{t-1}$. The output-independence assumption states that the probability that a symbol is emitted at time t depends only on the state s_t and is independent of the past observation.

Given the definition of an HMM above, three basic problems have to be addressed in order to apply HMMs to speech applications.

- The Evaluation problem: Suppose an HMM is given, the task is to determine the probability that a particular sequence of the visible states was generated by that model. This problem can be solved using Forward or Backward algorithms [Dev85].
- The Decoding problem: Suppose an HMM and a set of observations are given. The task is to determine the most likely sequence of hidden states that led to those observations. This problem can be solved by using Viterbi algorithm [Vit67, FJ73].
- The Learning problem: For a given HMM $\lambda = (A, B, \pi)$ and set of training observations O, the task is to adjust these parameters that maximize the probability to observe O: $\lambda^* = \arg\max_\lambda P(O|\lambda)$. Baum-Welch method - a special case of expectation-maximization algorithms can solve this problem [DLR77].

For speech recognition, the emission probability distribution matrix B can be modeled by using Gaussian Mixture Models (GMM) or Deep Neural Networks (DNN) which are described in the next paragraphs.

Gaussian Mixture Model (GMM)

One of the most common techniques to model the emission probability of an HMM is the Gaussian Mixture Model. Each of the M components of the mixture model is a Gaussian probability density function. The likelihood for state

s_j is the weighted sum of all the mixture likelihoods.

$$b_j(x) = \sum_{m=1}^{M} c_{jm}(x|\mu^{(jm)}, \Sigma^{(jm)}) \tag{2.4}$$

where c_{jm} is the mixture weight for Gaussian m of state s_j. These priors should satisfy the standard constraints for a valid probability mass function:

$$\sum_{m=1}^{M} c_{jm} = 1, c_{jm} \geq 0 \tag{2.5}$$

Deep neural network (DNN)

Another approach to model the emission probability distribution is using a artificial neural network (ANN). An ANN/HMM hybrid model was first used for automatic speech recognition in 1990 (see [BM94]). This model was trained to predict the posterior probabilities of each HMM state. During decoding, the output probabilities were divided by the prior probability of each state to form a "pseudo-likelihood". However, the performance of the ANN/HMM could not outperform the GMM/HMM system since the complex structure was modeled by using only one hidden layer. Recent researches in Machine Learning have led to the development of algorithms which can be used to train deep neural networks more efficiently ([HOT06, VLBM08]). One of these approaches is the Deep Belief Network (DBN), a multi-layered generative model which can be trained greedily, layer by layer using Restricted Boltzmann Machine at each layer ([HOT06]). It has been observed that using parameters of a DBN to initialize a deep neural network (DNN) - a neural network with many hidden layers - before fine tuning with backpropagation leads to a better performance of a DNN. This idea has been recently applied to the ANN/HMM hybrid system [SLY11, DYDA12, MDH12] and led to a significant improvement in different tasks with different data sets.

Restricted Boltzmann Machine (RBM) are bipartite undirected graphical models, with a set of nodes corresponding to observed random variables (also called visible units, v) and a set of nodes corresponding to latent random variables (or hidden units, h), that only allow interactions between the two sets of variables (that is, between the visible and hidden units) but not within each set of nodes. The joint probability of the visible units v and hidden units h is defined as:

$$P(v, h) = \frac{1}{Z_{h,v}} e^{E(v,h)} \tag{2.6}$$

where $Z_{h,v}$ is the normalizing partition function. Visible units are real-valued for speech observations and binary-valued otherwise and hidden units are always binary-valued. In the case of binary visible units, a Bernoulli-Bernoulli RBM can be used. Its energy function is:

$$E_{B_B}(v,h) = -v^T W h - b^T v - a^T h \tag{2.7}$$

For real-valued visible units, a diagonal covariance Gaussian-Bernoulli RBM is used. Its energy function is given by:

$$E_{G_B}(v,h) = -v^T W h - \frac{1}{2}(v-b)^T(v-b) - a^T h \tag{2.8}$$

W is a symmetric weight matrix defining interactions between vectors v and h while b and a are additive bias terms. RBM pre-training maximizes the likelihood of the training samples using the contrastive divergence algorithm [HOT06]. If many layers have to be initialized, the parameters of the given layer are fixed and its output is used as the input to the higher layer which is optimized as a new RBM. This can be repeated as many times as desired to produce many layers of non linear feature detectors that represent progressively more complex structure in the data. The RBMs can be combined to produce a single, multilayer generative model called Deep Belief Network (DBN).

DNN acoustic model training: Finally, the generative weights can simply be used in the reverse directions as a way of initializing all the feature detecting layers of a feed-forward neural network. Then, the final softmax layer can be added and fine-tuning using error back propagation (BP) [RHW02a] can be performed discriminatively.

DNN initialization: After the success of the results of [SLY11, DYDA12, MDH12], many research works were performed in this direction and earned a lot of attention in the speech community. One of the main challenges of the DNN training is initialization. Using pre-trained DBN is one of several initialization methods. The traditional way is to initialize the DNN parameters with random values, for example in a specified interval. Furthermore, another method called "discriminative pre-training" which has been proposed in [SLCY11] could be applied. In this approach, a one-hidden-layer DNN is trained to full convergence first. For this, senone labels with BP are used. Then, the softmax layer is replaced by another randomly initialized hidden layer and a new random softmax layer is added on top of this. Afterwards, the network is discriminatively trained again until full convergence. This process is repeated until the desired number of hidden layers is reached. In [SLCY11], it was shown that there is no significant difference in terms of performance between using pre-trained DBN and discriminative pre-training techniques. Moreover, using discriminative pre-training is even slightly better than using pre-trained DBN when the number of hidden layers increases.

Acoustic modeling unit

The acoustic modeling unit is the first important question which should be carefully explored to build an ASR system. [HAH01] mentioned that an *accurate*, *trainable*, and *generalizable* unit should be used. That means,

- The unit should represent the acoustic realization that appears in different contexts (*accurate*).
- Enough training data should be available to train the parameters of the unit (*trainable*).
- It should be possible to derive new words from a predefined unit inventory (*generalizable*).

Obviously, the word unit is accurate but not trainable and generalizable for LVSCR. Therefore, we concentrate on discussing smaller unit, such as phones, sub-phones, and context-depedent subphones.

Phones/Subphones Compared to word units, phones are a better choice for LVSCR. Most of the languages have less than 50 phones and, therefore, the acoustic model for those phones can be trained with a reasonable amount of data. Moreover, they are vocabulary independent and can be trained on one task and tested on another. To model a phone, 3-states HMM is typically used. It means that a phone is divided into three subphones: the begin, the middle and the end of the phone. The most important reasons are 1) a phone sounds different at the beginning, in the middle or at the end, and 2) the minimum duration of a phone is around 30ms, which corresponds to at least three HMM states since each state emits at least one frame of 10ms length. However, the phonetic model is inadequate because it assumes that a phone is identical in different contexts. Due to co-articulation effects, the phones in a word are not produced independently. Thus, the realization of a phone is strongly affected by its neighboring phones.

Context dependent phones/subphones One of the most important techniques which is widely used for acoustic modeling is context-dependent modeling [Lee88]. Started with the motivation, that phones sound differently depending on the preceding and the following phones due to coarticulation effects, different acoustic models are trained for a phone dependent on the context of this phone. In general, a context dependent phone is known as *polyphone*. However, depending on the width of the context, different terms, such as triphone (one left and one right context) or quintphone (two left and two right contexts) are defined. The most popular technique used to cluster the context-dependent phones is using decision trees [LHH$^+$90]. It allows finding acoustic models for all context dependent phones even if they do not appear in the training data. Moreover,

in [HHA96] it was shown that applying clustering on subphone level is better than on phone level. Therefore, context-dependent subphones - known as *senones* - have become state-of-the-art techniques for context dependency modeling of LVCSR. The questions of the tree could be chosen based on linguistic knowledge or data-driven. Figure 2.5 illustrates an example of a context decision tree. In this case, the questions are defined using linguistic knowledge, e.g. is the left context of the phone a vowel? Is the right context of the phone a fricative?

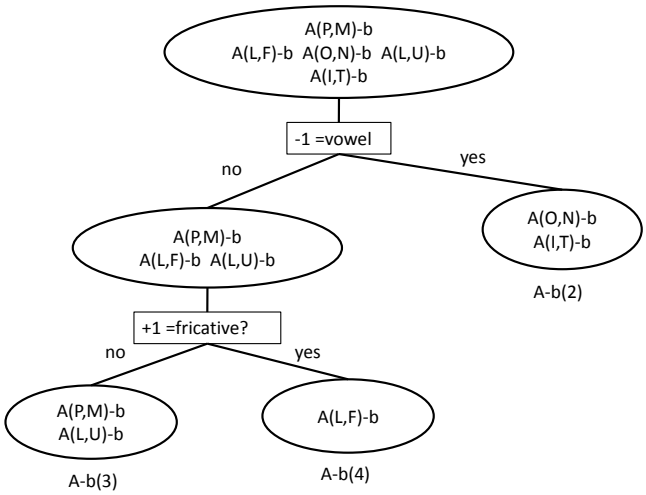


Figure 2.5: Context dependent decision tree for the phone state A-b

2.2.3 Language modeling

This section provides a short overview of three different kinds of language models (LMs): N-gram, Factored language model (FLM), and Recurrent Neural Network language model (RNNLM). N-gram is the traditional technique which is mainly used in many speech related applications. FLM [BK03] and RNNLM [MKB$^+$10] are advanced techniques which earned a lot of attention in the speech processing community since they provide substantial improvements over the N-gram in many tasks on different databases. They also allow

to easily integrate additional linguistic features to obtain better and more robust language models.

N-gram language model

The N-gram language model used in speech recognition captures automatically extracted linguistic knowledge about the target language from text. It helps to select the best option for a word transition. Language and acoustic models are computed separately and, then, connected as illustrated in equation 2.1 to help the search algorithm to find the most likely word sequence. The N-gram model can be computed from a text corpus. It is a process of counting the occurrences of a given word W in some history H. The history contains the previous $n-1$ words from a text and depending on n, the LM can be unigram (no history considered), bigram (a context of 2 words, i.e. history of one word considered), trigram, etc. The probability of a given sequence of words can be computed using trigram language models with the help of following equation:

$$P(w_{n-2}w_{n-1}w_n) = P(w_{n-2})P(w_{n-1}|w_{n-2})P(w_n|w_{n-1}w_{n-2}) \tag{2.9}$$

To estimate the N-gram probabilities for trigrams, the occurrences of w_{n-2}, w_{n-1}, w_n and w_{n-2}, w_{n-1} are counted in a training text. Afterwards, $P(w_n|w_{n-1}, w_{n-2})$ can be computed using the following equation:

$$P(w_n|w_{n-1}w_{n-2}) = \frac{C(w_{n-2}, w_{n-1}, w_n)}{C(w_{n-2}, w_{n-1})} \tag{2.10}$$

The main challenge of this training procedure is data sparseness. For example, if a bigram never occurs in the training data, its probability would be 0. Therefore, if the bigram appears in a sentence of the testing data, the probability for the whole sentence would be 0. This is an obvious underestimation of those sentences. To escape the problem of assigning a zero probability to a phrase that actually can occur as valid language construct but did not occur in the training text, different LM smoothing techniques can be applied. The strategies used to implement LM smoothing are discounting, back-off and interpolation with lower order models. Discounting techniques subtract a defined number from the counts of frequently occurring n-grams and distribute it to the n-grams that do not occur frequently. Another way to smooth the probability distributions of the n-grams is to back off to lower order models. If a given n-gram does not occur in the training data, usually the $n-1$-gram distribution is used.

Factored language model (FLM)

In a factored language model [BK03], a word is regarded as a vector of n factors, hence $w_t = f_t^1, f_t^2, .., f_t^n$. Factors can be, for example, morphological classes, stems, roots, and other features. In highly inflected languages (e.g., Arabic, German, Finnish), morphological features may be helpful, while for sparsely inflected languages, data-driven word classes or semantic features may provide useful information. Obviously, the standard N-gram language models are special cases of FLMs, since the factors could be the words themselves. If a sequence of features has not been detected in the training data, back-off will be used. Unfortunately, the number of possible parameters is rather high: Different feature combinations from different time steps can be used to predict the next word (conditioning factors). Furthermore, different back-off paths and different smoothing methods may be applied. To detect useful parameters, the genetic algorithm described in [DK04] can be used. It is an evolution-inspired technique that encodes the parameters of an FLM as binary strings (genes). First, an initializing set of genes is generated. Then, a loop follows that evaluates the fitness of the genes and mutates them until their average fitness does not improve any more. As fitness value, the inverse perplexity of the FLM corresponding to the gene on the development set is used. Hence, parameter solutions with lower perplexities are preferred in the selection of the genes for the following iteration. In [DK04], it is shown that this genetic method outperforms both knowledge-based and randomized choices. An example of a back-off graph is illustrated in figure 2.6. In this example, part-of-speech (POS) tags and words are used as features. The three conditioning factors contain the previous word W_{t-1} and the two previous POS tags P_{t-1} and P_{t-2}.

Recurrent neural network language model (RNNLM)

Another option to estimate the probability of a word given a specific context is using a recurrent neural network [MKB+10]. Figure 2.7 illustrates the idea of this model. Vector $w(t)$ forms the input of the recurrent neural network. It represents the current word using 1-of-N coding. Thus, its dimension equals the size of the vocabulary. Vector $s(t)$ contains the state of the network. It is called 'hidden layer'. The network is trained using back-propagation through time (BPTT) [Wer90], an extension of the back-propagation algorithm for recurrent neural networks. With BPTT, the error is propagated through recurrent connections back in time for a specific number of time steps t. Hence, the network is able to remember information for several time steps. The matrices U, V and W contain the weights for the connections between the layers. These weights are learned during the training phase. Moreover, the output layer is factorized into classes to accelerate the training and testing processes. Every word belongs to

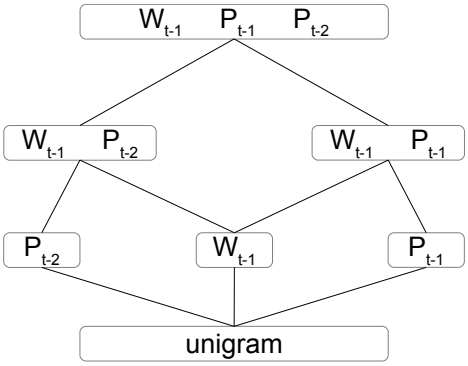


Figure 2.6: Possible back-off graph for a FLM using the previous word W_{t-1} and the part-of-speech tags of the last two previous words P_{t-2}, P_{t-1} as features

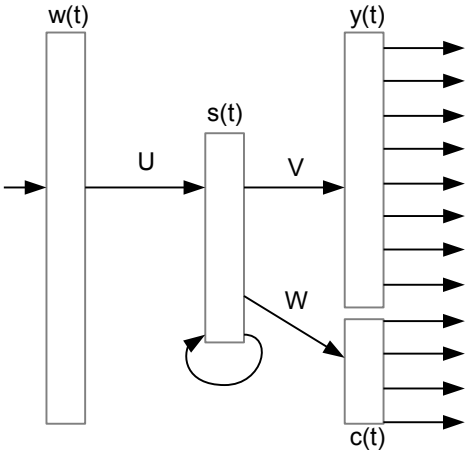


Figure 2.7: Recurrent neural language model [MKB+10]

exactly one class. The classes are formed during the training phase depending on the frequencies of the words. Vector $c(t)$ contains the probabilities for each class and vector $y(t)$ provides the probabilities for each word given its class.

Hence, the probability $P(w_i|history)$ is computed as shown in equation 2.11.

$$P(w_i|history) = P(c_i|s(t))P(w_i|c_i, s(t)) \tag{2.11}$$

This technique has several advantages over the N-gram language model since it can handle a very long history context. Furthermore for every word of the output, a probability could be obtained. Hence, this model captures smoothing implicitly.

2.2.4 Combining acoustic and language models

According to Bayes's equation 2.1, the acoustic model probability and the language model probability can be combined through simple multiplication. In practice, we need to add a *language model weight* and an *insertion penalty*.

The acoustic model probability is usually underestimated due to the Markov fallacy and the independence assumptions [HAH01]. Therefore, combining the language model probability with the underestimated acoustic model probability would give the language model too little weight. Moreover, the two quantities have a different range if continuous HMMs are used. With a *language model weight* LW, the LM probability $P(W)$ becomes $P(W)^{LW}$.

Furthermore, a penalty for inserting a new word is introduced. If the penalty is large, the decoder will prefer fewer longer words in general and vice versa. To adjust the penalty of inserting new words, the *insertion penalty* IP is used. Therefore, the language model contribution becomes:

$$P(W)^{LW} IP^{N(W)} \tag{2.12}$$

where $N(W)$ is the number of words in the sentence W. Both LW and IP are typically determined empirically to optimize the ASR performance on a development set.

2.2.5 N-best lists and word lattices

The output of an ASR system is usually the first best hypothesis. However, for many applications, such as speech translation or information retrieval, it is common to store the top N best possible hypotheses. The lattice is a graph with connected word hypotheses in a time synchronous manner that represents the alternative hypotheses of the speech recognizer. Depending on the implementation, a word can be stored in a node or in an edge of the graph. If a node in

the word lattice represents the word hypothesis with the corresponding acoustic model score at the current time segment, then the language model scores can be stored as transition probabilities on the word lattice links.

From the lattice, it is possible to extract the N-best hypotheses. One of the most widely used techniques which is used to extract the N-best list from a word lattice is presented in [SKW97a]. Obviously, every N-best list is only a part of the lattice. This also means that information is lost when N-best lists instead of lattices are used. However in scenarios with limited processing time, N-bests lists can be very helpful.

2.2.6 Unsupervised training of acoustic models

One of the main challenges of standard acoustic model training is the need for transcriptions and high costs to create transcriptions respectively. In 1998, Zavaliagkos and Colthurst started the first explorations towards unsupervised training to improve ASR performance [ZC98]. The idea is to use an existing speech recognizer to generate automatic transcriptions for available untranscribed audio data. With confidence measures derived from the recognizer output, the hypotheses which have a higher confidence score than a specified threshold are selected as transcriptions. This threshold is normally a design parameter to control the tradeoff between the amount of selected data and the quality of the automatic transcriptions. The selection of appropriate transcriptions is crucial to the resulting recognizer performance.

Confidence measures The confidence of a speech recognizer output expresses the certainty of the emitted hypothesis. The less confusion exists while generating the output hypothesis, the more confident the system is. However, high confidence does not always correlate with a correct hypothesis. Therefore, confidences have to be treated carefully, especially, if the speech recognizer has a high overall word error rate. Confidence can be measured at different levels of the recognizer output: utterance based confidence measures indicate the certainty regarding a whole sentence and also its semantic context; word based confidence measures abstract from the semantic context and give a confidence score for each word in an utterance; phone level and frame level confidence measures provide a more precise indication of the certainty of the acoustic model, apart from any semantic. In this thesis, two different confidence measures based on word lattices are used, namely gamma and A-stabil [SK97].

Gamma corresponds to the link probability in the word lattice. A node in the word lattice represents an HMM state and is associated with a word in the hypothesis whereas the emission probability of each HMM state corresponds to

the acoustic model score of this word at the current time segment. The transition probabilities of the word lattice links represent the language model scores. Given the emission probabilities and the transition probabilities of the word lattice, the link probability can be computed with the standard forward-backward algorithm.

A-stabil refers to acoustic stability and is computed at a higher level of the word lattice. A fixed number (typically 100) of different hypotheses are produced using the lattice. Each of those hypotheses results from a different weighting between acoustic model and language model. It is then aligned against a reference output of the recognizer, which is defined as the supposedly best hypothesis. For each word in the reference output, the number of occurrences in the alternative hypotheses is counted and divided by the total number of alternative hypotheses. The result of this calculation serves as confidence score for this word.

The quality of confidence measures usually depends on the recognizer. If the recognizer performance is good, the confidences are more reliable. Whereas confidences produced by a recognizer with poor performance tend to be unreliable. That means even though a word has a very high confidence score, the word itself may be wrong nonetheless. In order to maintain reliability of confidences, a threshold for the least reliable confidence has to be selected carefully.

2.2.7 Acoustic model adaptation

Acoustic model adaptation is a technique used to modify the acoustic models of a speech recognizer to better match specific speakers or conditions. It is widely used in many speech recognition systems to improve the performance for the user. Adaptation can transform a speaker-independent system into a speaker-dependent one. If not enough data is present to train a real speaker dependent system, general models can be used as a starting point. The idea of adaptation is using a small amount of specific speech data to calibrate the already trained general models towards the new conditions. Hence, adaptation is a powerful concept inspired by methods which humans use to understand speech with never seen properties. There are different techniques of acoustic model adaptation. Batch adaptation, for example, means adapting the system in one step with all the adaptation data. Another possibility is incremental adaptation which runs the adaptation process in the background and adapts while the user is speaking. Generally, adaptation can also be categorized according to the available transcriptions as supervised or unsupervised adaptation.

This section describes two widely used acoustic model adaptation techniques called Maximum Likelihood Linear Regression (MLLR) and Maximum a Posteriori (MAP).

Maximum Likelihood Linear Regression (MLLR)

Maximum Likelihood Linear Regression (MLLR) is a method that transforms the parameters of the emission Gaussian density functions of an HMM in a linear manner. This kind of transformation captures linear relationships between the general models and the adaptation data. The transformation can be applied either in the model space or in the feature space. When using MLLR adaptation, either exclusively the means or additionally the variances of the Gaussian distributions are transformed [Gal98, LW95]. It is also possible to decouple the means from the variances and transform them separately which is defined as unconstrained MLLR in [Gal98].

$$\begin{aligned} \tilde{\mu_{sm}} &= A_s \mu_m \\ \tilde{\Sigma_{sm}} &= H_s \Sigma_m H_s^T \end{aligned} \tag{2.13}$$

If the two matrix transformations are constrained to be the same, then a linear transform related to a feature space transform can be obtained. This is called contrained MLLR [LW95] :

$$\begin{aligned} \tilde{\mu_{sm}} &= \tilde{A}_s \mu_m \\ \tilde{\Sigma_{sm}} &= \tilde{A}_s \Sigma_m \tilde{A}_s^T \end{aligned} \tag{2.14}$$

Parameter estimation The transformation matrix is estimated to maximize the likelihood given the adaptation data in supervised or unsupervised mode. For supervised adaptation, the transcription is known and can be directly used without further consideration. If used in unsupervised mode, the transcriptions must be derived from the recognizer output. In this case, MLLR is normally used iteratively to increase the transcription quality and, therefore, the adaptation process. The confidence score can be used to weight the automatic transcription.

Regression class tree If adaptation data is limited, the transformation can be shared across different Gaussians in the system. The number of transformations to use for any specific set of adaptation data can be determined automatically using a regression class tree in figure 2.8. Each node represents a regression class, i.e. a set of Gaussian components which will share a single

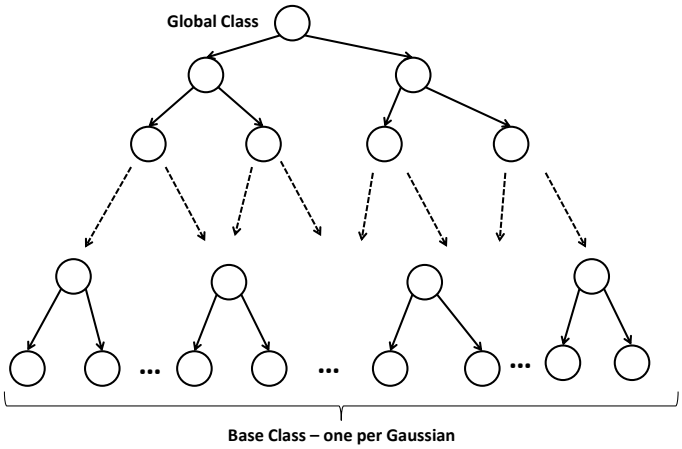


Figure 2.8: A regression class tree

transform. The total occupation count associated with any node in the tree can be easily computed since the counts are known at the leaf nodes. Then for a given set of adaptation data, the tree is descended and the most specific set of nodes is selected for which there is sufficient data. This regression tree can be automatically trained by applying a clustering technique to categorize Gaussians.

Maximum a Posteriori (MAP)

The Maximum a Posteriori (MAP) adaptation [GL94] tries to re-estimate the HMM parameters given an observed signal. Let $\lambda_{ik} = (\mu_{ik}, \Sigma_{ik})$ be the k-th Gaussian component of state i with corresponding mixture weight ω_k. Given the observation samples $X = (x_1, ..., x_M)$, the update equation for the mean vector can be formulated as followed:

$$\tilde{\mu_{ik}} = \frac{\tau_{ik}\mu_{ik} + \sum c_{ikt}x_t}{\tau_{ik} + \alpha \sum c_{ikt}} \tag{2.15}$$

where

$$c_{ikt} = \frac{\omega_k N(x_t|\lambda_{ik})}{\sum \omega_k N(x_t|\lambda_{ik})} \tag{2.16}$$

and $\tilde{\mu_{ik}}$, μ_{ik} denote the initial and the adapted vector respectively. τ is normally determined empirically. Since every Gaussian component is updated individually, MAP adaptation suits well to the case that enough adaptation data is available.

2.2.8 Evaluation criteria

Language model performance

To evaluate a language model, the out-of-vocabulary (OOV) rate and perplexity (PPL) can be computed on a test set. The OOV rate gives the number of tokens in a test set which are not covered by the vocabulary of the language model. The perplexity of a language model is derived from the entropy $H(W)$ of the test sequence. It can be computed using the following equation.

$$H(W) = -\sum P(W) log P(W) \tag{2.17}$$

The perplexity is then obtained as $2^{H(W)}$. For a fixed OOV, language models with lower perplexity are usually sought, although it is known that the perplexity is only loosely correlated with the performance of an ASR system.

ASR performance

The standard metric to evaluate an ASR system is the word error rate (WER). The output of the decoding process is a hypothesis for what has been spoken. Comparing the hypothesis with the reference text which is the true transcription of what has been said, yields a score in the form of the percentage of errors made. The following errors can occur after the alignment of the hypothesis and the reference text:

- Substitution: a word is misrecognized
- Deletion: a word from the reference is missing in the hypothesis
- Insertion: the recognizer inserts a word that has actually not been spoken

To compute the WER after identifying these errors, the following equation is used:

$$WER[\%] = \frac{\#substitutions + \#insertions + \#deletions}{\#words(reference)} * 100\% \tag{2.18}$$

The equation above shows that the WER can exceed 100%, especially in the case that the speech recognizers tends to insert words. The word error rate can be transformed into different similar measurements, such as character error rate or syllable error rate depending on the language. For a special task like recognizing Mandarin English Code-Switching speech, word error rates can be applied for English and character error rates for Mandarin respectively. Therefore, the measurement is called mixed error rate (MER). The presented MER is the weighted average over all English and Mandarin portions of the speech recognition output. By applying character based error rates for Mandarin, the performance does not dependent on the applied word segmentation algorithm for Mandarin and, thus, performance can be compared across different segmentations, providing more flexibility for future investigations.

CHAPTER 3

Data, Tools and Baseline (ASR) Systems for Multiple Languages

This chapter briefly reviews all the databases which are used in the thesis including the GlobalPhone database, the accented speech corpus and the SEAME corpus. Afterwards, the monolingual speech recognition systems which serve as baseline for many experiments in this thesis and their performance on the GlobalPhone database are presented.

3.1 Data corpora

3.1.1 GlobalPhone database

GlobalPhone is a multilingual data corpus developed at Karlsruhe Institute of Technology (KIT) [Sch02, SVS13]. The complete data corpus comprises (1) audio/speech data, i.e. high-quality recordings of spoken utterances read by native speakers, (2) corresponding transcriptions, (3) pronunciation dictionaries covering the vocabulary of the transcripts, and (4) baseline N-gram language models. The first two parts are referred to as GlobalPhone Speech and Text

Database (GP-ST), the third as GlobalPhone Dictionaries (GP-Dict), and the latter as GlobalPhone Language Models (GP-LM). GP-ST is distributed under research or commercial license by two authorized distributors, the European Language Resources Association (ELRA) [ELR12] and Appen Butler Hill Pty Ltd. [htt12]. GP-Dict is distributed by ELRA, while the GP-LMs are freely available for download from our website [LB12].

Language Coverage

To date, the GlobalPhone corpus covers 20 languages, namely Modern Standard Arabic (AR), Bulgarian (BG), Chinese-Mandarin (MA), Chinese-Shanghai (SH), Croatian (HR), Czech (CZ), French (FR), German (GE), Hausa (HA), Japanese (JA), Korean (KR), Polish (PL), Brazilian - Portuguese (PT), Russian (RU), Latin American - Spanish (SP), Swedish (SW), Tamil (TA), Thai (TH), Turkish (TU), and Vietnamese (VN). This selection covers a broad variety of language peculiarities relevant for Speech and Language research and development. It comprises wide-spread languages (e.g. Arabic, Chinese, Spanish, Russian), contains economically and politically important languages, and spans wide geographical areas (Europe, Africa, America, Asia). The spoken speech covers a broad selection of phonetic characteristics, e.g. tonal sounds (Mandarin, Shanghai, Thai, Vietnamese), pharyngeal sounds (Arabic), consonantal clusters (German), nasals (French, Portuguese), and palatized sounds (Russian). The written language contains all types of writing systems, i.e. logographic scripts (Chinese Hanzi and Japanese Kanji), phonographic segmental scripts (Roman, Cyrillic), phonographic consonantal scripts (Arabic), phonographic syllabic scripts (Japanese Kana, Thai), and phonographic featural scripts (Korean Hangul). The languages cover many morphological variations, e.g. agglutinative languages (Turkish, Korean), compounding languages (German), and also include scripts that completely lack word segmentation (Chinese, Thai, Vietnamese).

Data Acquisition

The data acquisition was performed in countries where the language is officially spoken. In each language about 100 adult native speakers were asked to read about 100 sentences. The read texts were selected from national newspaper articles available from the web to cover a wide domain with large vocabulary. The articles report national and international political news, as well as economic news, which makes it possible to compare the usage of proper names (politicians, companies, etc.) across languages. The following newspapers were used: Assabah for Arabic, Banker, Cash, and Sega for Bulgarian, Peoples

Daily for Mandarin and Shanghai Chinese, HRT and Obzor Nacional for Croatian, Ceskomoravsky Profit Journal and Lidove Noviny newspaper for Czech, Le Monde for French, Frankfurter Allgemeine und Sueddeutsche Zeitung for German, CRI online and RFI for Hausa, Hankyoreh Daily News for Korean, Nikkei Shinbun for Japanese, Folha de Sao Paulo for Portuguese, Dziennik Polski for Polish, Ogonyok Gaseta and express-chronika for Russian, La Nacion for Spanish, Goeteborgs-Posten for Swedish, Thinaboomi Tamil Daily for Tamil, Bangkok Biz news and Daily News for Thai, Zaman for Turkish, and Tin Tuc among others for Vietnamese. The speech data was recorded with a close-speaking microphone and is available in identical characteristics for all the languages: PCM encoding, mono quality, 16bit quantization, and 16kHz sampling rate. Most recordings were peformed in ordinary rooms, in the majority without background noise, so that the speakers were not distracted. The quality of noise level and recording room setup was reported for each session. The speakers were given instructions about the equipment handling in advance. They were introduced to the project goals and were allowed to read the texts before recording. The transcriptions are available in the original script of the corresponding language. In addition, all transcriptions have been romanized, i.e. transformed into Roman script applying reversible 1:1 character mappings. The transcripts were internally validated and supplemented by special markers for spontaneous effects like stuttering, false starts, and non-verbal effects, such as breathing, laughing, and hesitations. Speaker information, such as age, gender, place of birth, dialect, occupation, etc. as well as information about the recording setup complement the database.

Corpus Statistics

The entire GlobalPhone corpus contains over 400 hours of speech spoken by more than 1900 native adult speakers. The data are organized by languages and speakers and are divided in speaker disjoint sets for training (80%), development (10%), and evaluation (10%). Research work in this thesis used data from 15 different languages, namely Bulgarian, Czech, French, German, Hausa, Croatian, Japanese, Korean, Mandarin, Polish, Russian, Spanish, Tamil, Thai, and Vietnamese. Table 3.1 summarizes the amount of transcribed speech data of these relevant languages.

GlobalPhone Pronunciation Dictionaries

Phone-based pronunciation dictionaries are available for each GlobalPhone language. The dictionaries cover the words which appear in the transcriptions. The majority of the dictionaries were constructed in a rule-based manner using

Table 3.1: *GlobalPhone Corpus Statistics*

Language	Training [hrs:min]	Development [hrs:min]	Evaluation [hrs:min]
Bulgarian	16:47	2:16	1:56
Czech	26:49	2:22	2:41
French	24:55	-	2:01
German	14:54	1:57	1:28
Hausa	6:36	1:02	1:06
Croatian	11:48	2:02	1:45
Japanese	21:51	1:26	1:40
Korean	16:34	2:09	2:04
Mandarin	26:38	1:59	2:25
Polish	18:39	2:47	2:16
Russian	21:08	2:41	2:36
Spanish	17:35	1:40	2:03
Tamil	15:50	1:04	1:00
Thai	19:05	2:03	1:58
Vietnamese	22:15	1:40	1:30

language specific phone sets. After this automatic creation process the dictionary was manually post-processed by a native speaker, correcting errors in the automatic pronunciation generation and introducing pronunciation variants. To enable the development of multilingual speech processing, the phone names are consistent across languages, leveraging the International Phonetic Alphabet (IPA) [Ass99]. Table 3.2 gives an overview of the size of the phone sets, amount of vocabulary words covered, and amount of pronunciation variants of the 15 selected languages in the GlobalPhone pronunciation dictionaries.

3.1.2 Non-native speech database

To conduct experiments with non-native speech, an accented database was collected as an extension of the GlobalPhone database, named *GlobalPhone Accented* (GPA). Until today, *GPA* contains English with four different non-native accents [Mih11] and German with Chinese accent [Wan13].

Table 3.2: *GlobalPhone Pronunciation Dictionaries*

Languages	#Phones	#Words	#Dict entries
Bulgarian	44	275k	275k
Czech	41	277k	277k
French	39	122k	195k
German	43	39k	41k
Hausa	33	43k	48k
Croatian	32	21k	23k
Japanese	31	9k	13k
Korean	39	1.3k	3k
Mandarin	49	73k	73k
Polish	36	34k	34k
Russian	47	39k	40k
Spanish	42	31k	39k
Tamil	41	288k	292k
Thai	44	23k	25k
Vietnamese	38	30k	39k

English with non-native accents

In [Mih11], 63 non-native speakers of English (approximately 10 hours) were recorded. Table 3.3 presents some statistic about this corpus. Since there are many differences between the accents of people with various language backgrounds, this research is focused on four major groups of speakers: Native speakers of Bulgarian (BG), Chinese (Mandarin or Cantonese) (CH), German (GE) and some of the languages spoken in India (Hindi, Marathi, Bengali, Telugu, Tamil) (IN). The choice of these speaker groups was based on the availability of subjects as well as on the fact that these languages stem from different language families. Bulgarian is from the Slavic language family, Mandarin and Cantonese are members of the Sino-Tibetan language family, German is a Germanic language and the Indian languages belong to several language families, such as the Indo-European or the Dravidian language family. The recorded read speech sentences are extracted from the Wall Street Journal database [PB92]. The majority of topics are economy related news. All subjects were asked to read approximately 30 English sentences unique for each speaker within an accent and 6 sentences that are the same for everyone.

Depending on the speaker's self confidence and experience with the language, the recording of the sentences took between 30 minutes and an hour.

English with German and Bulgarian accent was recorded in Germany, while

Table 3.3: *GlobalPhone Accented Corpus Statistics*

	Total	BG	CH	GE	IN
#speakers	63	16	17	15	15
male/female	42/21	9/7	11/6	10/5	12/3
audio length [min]	490	125	149	107	109
time/speaker [min]	7.47	7.46	8.42	7.8	7.14
#tokens	57.4	14.3k	15.8k	13.6k	13.9k
#tokens/speaker	911	890	927	904	924
#utterances	2,368	583	640	565	580
#utts/speaker	37	36	37	37	38

the speech data for the Chinese and Indian databases were collected in the USA. The speakers from India have spent two years in average as residents in the USA, the Chinese speakers approximately 2.5 years. The numbers for the German and the Bulgarian speakers are 4.5 months and less than a month, respectively. All the speakers are at an age between 21 and 30: BG (21 - 29), CH (22 - 30), GER (22 - 30), IN (21 - 29). All the recordings were performed in a quiet room.

The division of the speakers that is used for the experiments is as follows: 5 speakers from each accent form the test set, 5 speakers are in the development set and additional 5 speakers from each accent are used for the acoustic model adaptation experiments or to train a new system. As the read text is taken from the Global Phone Database, the utterances are also available in native speech. Five speakers from every test or development set read the utterances of 10 speakers from the English Global Phone database, which means that two native speakers map to one non-native speaker from each accented database.

German with Chinese accent

To conduct crosslingual accent adaptation experiments, we collected about three hours German speech with Chinese accent [Wan13]. Chinese students at Karlsruhe Institute of Technology were asked to read about 50 German sentences selected from the German GlobalPhone database in a relative quiet room. The recordings took between 30min and 70min per person. In total, the corpus contains 21 speakers whose ages are between 19 and 32. Their native language is Mandarin. Most of the speakers have spent less than one year as residents in Germany. Table 3.4 presents some statistical information about the adaptation, development and testing data.

Table 3.4: *German with Chinese accent speech corpus statistics*

	Total	Adaptation	Dev	Eval
#speakers	21	9	6	6
male/female	12/9	5/4	3/3	4/2
audio length [min]	186	75	52	59
time/speaker [min]	8,86	8,30	8,73	9,83
#utterances	1057	454	301	302

3.1.3 SEAME corpus

SEAME (South East Asia Mandarin-English) is a conversational Mandarin-English Code-Switching speech corpus recorded from Singaporean and Malaysian speakers, created and collected by [LTCL10]. The corpus was used for the research project 'Code-Switch' jointly performed by Nanyang Technological University (NTU) and Karlsruhe Institute of Technology (KIT). The recordings consist of spontaneously spoken interviews and conversations of about 63 hours of audio data. The corpus is designed for multiple research purposes which include language boundary detection, language identification studies and multilingual LVCSR systems. Hence, a word-level manual transcription with language boundary alignment is provided. As the corpus was developed for spontaneous Code-Switching speech research, the recordings consist of interviews and conversations without prepared transcriptions. Considering the particular speaking styles in Singapore and Malaysia, the transcribed words were classified into four categories for language identification research: English and Mandarin words, Silence and Others (discourse particles, other languages, and hesitations). The ratio in tokens of Mandarin, English, Silence and Others is 44%, 26%, 21% and 7% respectively. The average number of code switches within each utterance is 2.6 when counting only switches between Mandarin and English and ignoring the silence and others tags. In total, the corpus contains 9,210 unique English and 7,471 unique Mandarin words. The duration of monolingual segments is very short: More than 82% English and 73% Mandarin segments are less than 1 second long while the average duration of English and Mandarin segments is only 0.67 seconds and 0.81 seconds, respectively. We divided the corpus into three sets (training, development and test set) and distributed the data based on several criteria (e.g. gender, speaking style, ratio of Singaporean and Malaysian speakers, ratio of the four categories, and the duration in each set). Table 3.5 lists the statistics of the SEAME corpus in these three sets.

Table 3.5: *Statistics of the SEAME corpus*

	Train set	Dev set	Eval set
# Speakers	139	8	8
Duration(hours)	58.4	2.1	1.5
# Utterances.	48,040	1,943	1,015

3.2 Speech recognition for multiple languages

To conduct research in multilingual speech recognition, we developed monolingual ASR systems for the 15 languages using the GlobalPhone database. This section describes the acoustic and language models as well as the toolkits Janus and Rapid Language Adaptation Toolkit which were used to train the ASR system. Afterwards, we present some advanced techniques which we developed to optimize the ASR systems based on language peculiarities.

3.2.1 Acoustic modeling

Janus Speech Recognition Toolkit

To train acoustic models for multiple languages, we used the Janus speech recognition toolkit (JRTk) [FGH+97] which is a software developed at Carnegie Mellon University (CMU) and KIT. The toolkit includes an AM trainer which supports state-of-the-art AM training techniques and the dynamic decoder Ibis [SMFW01]. The AM training using Janus includes three main steps: context-independent AM training, decision tree building and context-dependent AM training. On top of that, speaker adaptive training or discriminative training based on boosted MMIE [PKK+08] can be applied. However, those two techniques are not used for the baseline systems described in this section since we aimed at developing speaker independent ASR systems. Furthermore, the amount of training data in the GlobalPhone database is rather small. Hence, discriminative training techniques may lead to no substantial improvements. In our experiments with Vietnamese, we observed less then 1% relative improvement over the baseline system. Since the discriminative training is CPU-intensive and time-consuming, we decided to not apply this technique on top of our baseline systems.

Acoustic model training

We used the multilingual inventory which has been trained earlier from seven GlobalPhone languages [SW01b] to bootstrap a system in a new language. First, an initial state alignment is produced by selecting the closest matching acoustic models from the multilingual inventory as seeds. The closest match is derived from an IPA-based phone mapping. The standard front-end is applied by using a Hamming window of 16ms length with a window overlap of 10ms. Each feature vector has 143 dimensions containing 13 Melscale Frequency Ceptral Coefficients (MFCC) and their five left and right neighbors. A Linear Discriminant Analysis transformation reduces the feature vector size to 42 dimensions. The acoustic model uses a fully-continuous 3-state left-to-right HMM. The emission probabilities are modeled by Gaussian Mixtures with diagonal covariances. For context-dependent acoustic models, we train a quintphone system and stop the decision tree splitting process at a specified language dependent threshold (varies between 500 and 3,000 leaves depending on the available amount of training data). After context clustering, a merge&split training [UNGH98] is applied, which selects the number of Gaussians according to the amount of data. For all the models, we used one global semi-tied covariance (STC) matrix [Gal99] after applying the Linear Discriminant Analysis (LDA) [Fuk90].

3.2.2 Language modeling

Rapid Language Adaptation Toolkit (RLAT)

The project SPICE (DARPA, 2004-2008) performed at the Language Technologies Institute at Carnegie Mellon and the Rapid Language Adaptation project at Cognitive Systems Lab (CSL) aims at bridging the gap between language and technology expertise. For this purpose, RLAT [RLA12] provides innovative methods and interactive web-based tools to enable users to develop speech processing models, to collect appropriate speech and text data to build these models, as well as to evaluate the results and improve the models iteratively [SBB+07]. The toolkit significantly reduces the amount of time and effort involved in building speech processing systems for unsupported languages. In particular, the toolkit allows the user to (1) design databases for new languages at low costs by enabling users to record appropriate speech data along with transcriptions, (2) to continuously harvest, normalize, and process massive amounts of text data from the web, (3) to select appropriate phone sets for new languages efficiently, (4) to create vocabulary lists, (5) to automatically generate pronunciation dictionaries, (6) to apply these resources by developing acoustic

and language models for speech recognition, (7) to develop models for text-to-speech synthesis, and (8) to finally integrate the built components into an application and evaluate the results using online speech recognition and synthesis in a talk-back function [SBB+07]. RLAT [RLA12] and SPICE are freely available online services which provide an interface to the web-based tools. They have been designed to accommodate all potential users, ranging from novices to experts. In this thesis, RLAT was applied to crawl the text material on the Internet which was then used to build the language models.

GlobalPhone Language Models

We applied RLAT to crawl a massive amount of text data and used the strategy presented in [VSKS10] to quickly and efficiently build the GlobalPhone language models for 18 languages. We crawled text data for several days, and each day one language model was built based on the daily crawled text data. The final language model was then created by a linear interpolation of all the daily language models. The interpolation weights were computed using the SRI Language Model Toolkit [Sto02], optimized on the GlobalPhone development sets. The experimental results in [VSKS10] indicated that the text data from the first few days are most helpful and, therefore, receive the highest interpolation weights in the final language model. Since the outcome of the crawling process depends on the input websites, the starting pages have to be chosen carefully. In our experiments, we found that in the case of Croatian, Japanese, Korean and Thai, the crawling process finished prematurely after one or two days, retrieving a rather small amount of text data. Since text data diversity has a major impact on language model quality and the final performance of an ASR system, we selected additional websites to harvest more diverse text data. The final best language models were then built based on the interpolation of the language models from a variety of websites. Table 3.6 gives an overview of the amount of crawled text data, the trigram perplexities (PPL), out-of-vocabulary (OOV) rates on the GlobalPhone test sets, and the vocabulary sizes of the language models for the 15 selected languages. For each language, the numbers of both the full (LM) and the pruned benchmark language models (LM-BM) are reported. The symbols in parentheses after the language name indicate the token units used, i.e. (w) for word-based, (s) for syllable-based, and (c) for character-based token units. The pruned benchmark language models are available for download in [LB12].

Table 3.6: *Text Resources and Language Models*

Language	3-gram PPL LM-BM	3-gram PPL LM	OOV [%]	#Vocab	#Tokens
Bulgarian (w)	454	351	1.0	274k	405M
Czech (w)	1421	1361	4.0	267k	508M
French (w)	324	284	2.4	65k	-
German (w)	672	555	0.3	38k	20M
Hausa (w)	97	77	0.5	41k	15M
Croatian (w)	721	647	3.6	362k	331M
Japanese (s)	89	76	1.0	67k	1600M
Korean (c)	25	18	0	1.3k	500M
Mandarin (c)	262	163	0.8	13k	900M
Polish (w)	951	904	0.8	243k	224M
Russian (w)	1310	1150	3.9	293k	334M
Spanish (w)	154	108	0.1	19k	12M
Tamil (s)	730	624	1.0	288k	91M
Thai (s)	70	65	0.1	22k	15M
Vietnamese (s)	218	176	0	30k	39M

3.2.3 Language specific system optimization

Depending on the language peculiarities, we applied different techniques to improve the ASR performance. The following paragraphs list several optimization techniques which we used to optimize the ASR performance depending on the languages.

Tonal languages To model tonal languages, such as Chinese, Hausa, Thai, and Vietnamese, we apply the "Data-driven tone modeling" approach, where all tonal variants of a phone share one base model [VS09, SDV+12]. The information about the tone is added to the dictionary in form of a tone tag. These tags are used as questions in the context decision tree when building context dependent acoustic models. This way, it is based on the data whether different tonal variants of the same basic phone are represented by different models or share the same basic phone model.

In the case of Vietnamese, we also experimented with integrating fundamental frequency information into the preprocessing step [VS09]. According to [Nol64], the cepstrum of a speech signal has a peak corresponding to the fundamental period which can be used to extract tone features. Therefore, we computed the cepstrum with a window length of 40ms and detected the position of the maximum of all cepstral coefficients starting with the 30th coefficient. Furthermore, the positions of the three left and right neighbors, and their first and second derivatives were considered. This resulted in 21 additional coefficients

(1 maximum, 3 left neighbors, 3 right neighbors plus the first and second order derivatives). These 21 coefficients were added to the original 143 dimensional feature vector. With an LDA transformation, we finally reduced the 164 dimensional feature vector to 42 dimensions. By using this technique, we obtained about 5% relative improvement on the Vietnamese test set [VS09].

Isolating languages In isolating languages like Vietnamese, the text data contains sequences of monosyllables, i.e. white spaces occur directly after each monosyllable and each monosyllable could be accepted as an individual word. Therefore, it is important to increase the history in the language model and the context width in the acoustic model to improve the ASR performance. Thus, we combined monosyllable words to multisyllable words by concatenating syllables using the method in [VS09]. For example, the Vietnamese multisyllable word "sinh1 vien1" (student) was merged from "sinh1" and "vien1". For this process, we had to overcome two challenges. First, we had to find suitable multisyllables. To solve this problem, we used a dictionary based approach and built a look up table to check whether the combination of monosyllables is a viable word. For the case of Vietnamese, we used an open source dictionary from the University of Leipzig [Dicb]. It contains about 23.000 bisyllable Vietnamese words and about 6.500 monosyllable words. The second problem was to figure out which syllables should be concatenated. Three methods have been described in the literature: apply statistical information, linguistic information, and a hybrid of both. To develop a language-independent technique, we relied on the statistical method. Using crawled text data, we calculated the frequencies of all bi-syllable words from the dictionary. For each sentence in the text corpus, we searched syllable by syllable for multisyllabic words from the beginning to the end of the sentence. Words with higher hit rate than the left and right neighbors were selected as multisyllabic words. With the resulting new text corpus we created a new language model with RLAT. Then, we concatenated the corresponding syllables in the transcriptions of the audio data and re-trained the acoustic model as well.

Morphological-rich languages Morphological-rich languages, such as Tamil may be a challenge for language models of state-of-the-art ASR systems. The morphological complexity often causes data sparsity problems and results in high OOV-rates and LM perplexities. A traditional approach to overcome this problem is to use a very large vocabulary. However, using a very large search vocabulary also leads to high OOV rates and high resource requirements, such as CPU time and memory. Alternatively, morpheme-based LMs can be used to lower the OOV rate, decrease the perplexity, reduce the resource requirements and achieve better accuracy. This paragraph presents the

technique called Dictionary Unit Merging Algorithm (DUMA) [KSW99, JVS12, JVS13] which is a data-driven, statistical approach to determine appropriate dictionary units. It should overcome the high OOV rate and LM perplexity due to the rich morphology of Tamil. The inputs of the algorithm are a pronunciation dictionary, the LM training text and a vowel list. The vowel list is the only linguistic knowledge required by the algorithm. Initially, the entire text was segmented into syllables which is language dependent. In the case of Tamil, the algorithm in [LM06] was applied. The word boundary information in the syllabified text was also included i.e. we inserted a "-" to every syllable that did not occur at the start of a word. Then, we obtained all possible syllable pairs from the syllabified text. Afterwards, each possible pair was looked up in the dictionary and the pronunciation of the vowel-vowel transition was retrieved. The merging algorithm is governed by the following iterative steps:

1. A hash table is computed that maps the vowel-vowel transition and the corresponding syllable pair to the frequency of the pair in the LM text.
2. For each vowel-vowel transition in the hash table, the most frequent syllable pairs are inserted into a merge-list.
3. All the pairs in the segmented corpus that can be detected in the merge-list are merged.

We only merged pairs that occur within a word, and chose not to merge pairs across word boundaries. We used the merge-list obtained after step 2 of the unit merging algorithm to merge both the training and test transcripts. Finally, we combined the units extracted by using this algorithm and the most frequent words to obtain the best ASR performance for Tamil [JVS13].

ASR performance

Figure 3.1 illustrates the ASR performance on the GlobalPhone test set for all the languages. Depending on the language, different kinds of error rates were used. Character error rate was applied for Korean, and Mandarin, while syllable error rate was used for Japanese, Tamil, Thai, and Vietnamese. The remaining languages were evaluated with word error rate. The ASR performance has a wide range from around 7.8% to 29.5% on the GlobalPhone test set.

We achieved error rates $< 15\%$ for Haitian Creole, Hausa, Mandarin, Polish, Spanish, Thai, and Vietnamese. Most of them served as baseline in our experiments which are described in the next chapters. For the case of Czech, we obtained two different baseline performances. The first baseline was used in Chapter 4 in which we assumed that the manual transcription of the audio

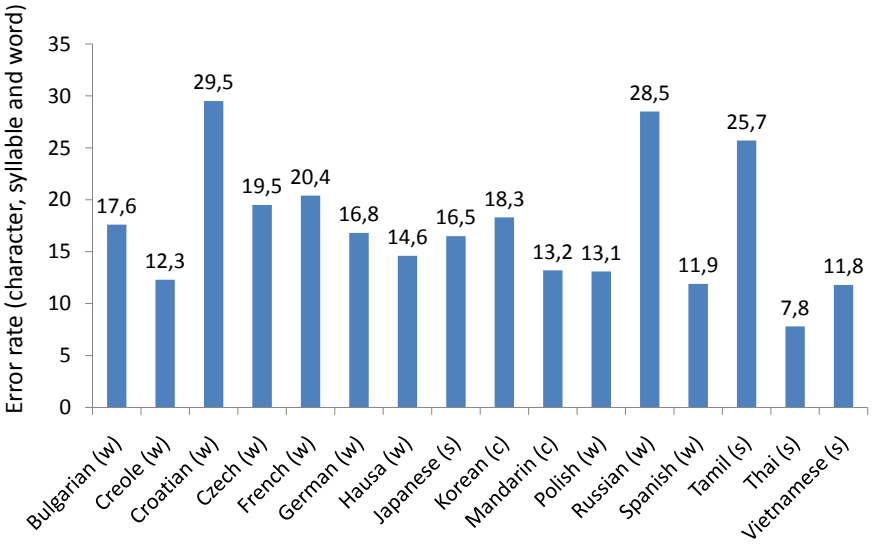


Figure 3.1: ASR performance on the GlobalPhone test set

data is not available. Therefore, we used a large decoding dictionary containing 267k words which covers the most frequent words of a web text corpus. On the Czech test set, we obtained 22.3% WER. The second baseline was used in Chapter 5 and 7 in which the manual transcriptions were used to select the most frequent 40k words as decoding dictionary entries. Using this smaller decoding dictionary, the WER was 19.5% on the Czech test set.

For morphological-rich languages, such as Bulgarian, Croatian, French, German, Korean, Russian, and Tamil, we obtained error rates larger than 20% for most of the cases. There could be three possible reasons for this. The first reason can lie within the difficulty of the language model task. To build an accurate LM with a large vocabulary, we need a large amount of text data. Even with many text data, the language model still has a high OOV rate and perplexity [VSKS10]. More specifically, in the case of Bulgarian, Czech, Croatian, Russian, Tamil, the search vocabulary is larger than 200,000 and, therefore, it leads to high perplexities of the language models on the test set. Another reason might be data inconsistencies including topic and domain because we

have been collecting the new text data since 2009 while many languages of the GlobalPhone database were recorded around 1998. Third, the results could be caused by special challenging aspects, such as *homophones* issues in French. These are words which have the same pronunciations and can, therefore, easily be confused.

CHAPTER 4

Cross-language Bootstrapping Based on Completely Unsupervised Training

With around 7000 languages in the world and the need to support multiple languages, the most important challenge nowadays is to port ASR systems to new languages rapidly and at reasonable costs. This chapter presents our multilingual unsupervised training framework which allows building an ASR system for new languages without any transcribed audio data - one of the most expensive and time-consuming steps when building an ASR system.

4.1 Introduction

Automatic speech recognition becomes more and more important in the daily life since it is used in many applications, such as dictation systems, navigation systems, speech translation systems and spoken web search. Due to the strong

growth of globalization, the need of ASR in many languages has increased dramatically over the last decade. One of the most challenging tasks is to minimize development costs and effort for the construction of a speech recognizer for a new language. Furthermore, large amounts of data have to be processed to allow speech recognition for continuously spoken speech. The principle that "there is no data like more data" [Jel05] is true in many contexts.

Modern media like the Internet provide a great amount of easily and freely accessible audio data for various languages. However, there are no restrictions in topic or vocabulary for these data, and one has to deal with different dialects or even different languages. Moreover, the most challenging problem with these data is the possible lack of transcriptions. Detailed transcriptions of audio training data are a crucial factor for the construction of automatic speech recognition systems. The generation of manual transcriptions requires 10 to 40 times real-time, depending, on the one hand, on the transcription quality and the transcriber's experience and, on the other hand, on the speaking style and also the quality of the audio data. Such effort is unbearable for the large amount of data that is nowadays used to build a recognizer for continuous speech. To overcome these problems and limitations, automatic methods to train a speech recognition system without transcribed audio data are required.

Moreover, many ASR systems for resource rich languages already exist. The question is whether we can use the knowledge and resources which are available to bootstrap systems for new languages. Figure 4.1 illustrates the initial situation: The available resources, such as pronunciation dictionary, audio data and text data of the new language are given, as well as many ASR systems in different languages. The goal is to build an ASR system for the new target language. In this work, we aim at developing a framework which allows building an ASR system for a new language using available resources with minimal human effort. In this scenario, we minimize the developing costs and time by automatically transcribing the audio data.

First, we revisit the cross-language transfer technique [SW01a] and investigate the impact of the relation between the source and target language on the ASR performance. Afterwards, different confidence scores, such as A-stabil and gamma are explored. We propose a new method to compute the word-based confidence score called "Multilingual A-stabil". Finally, we demonstrate that the proposed framework works well in different tasks with different databases even if the source and the target languages are not related.

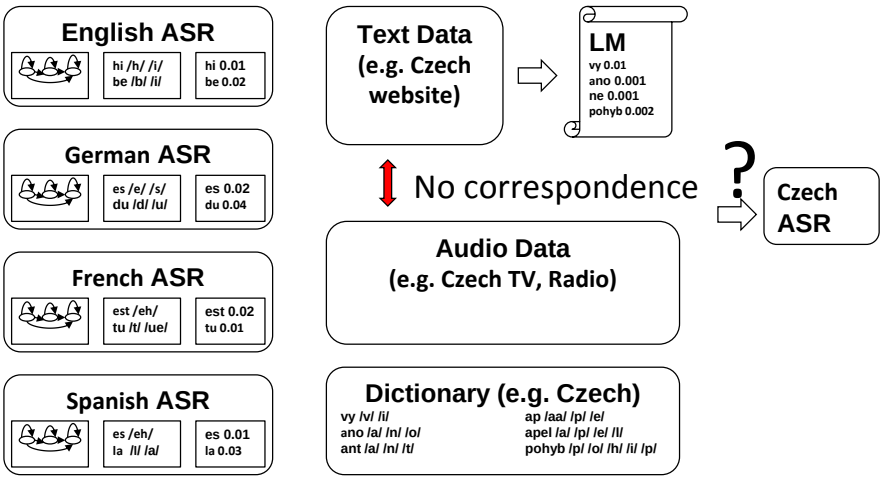


Figure 4.1: Initial situation: We assume to have pronunciation dictionaries and audio and text data of the new language (e.g. Czech) as well as several ASR systems of different languages (e.g. English, French, German, and Spanish). However, no transcriptions of the audio data are available.

4.2 Related work

4.2.1 Unsupervised and lightly unsupervised training

Unsupervised training in speech recognition showed its success in the past starting from 1998. The first explorations toward unsupervised training were conducted by Zavaliagkos and Colthurst [ZC98]. Afterwards, there were many studies, such as [KW99], [LGA02b], [LGA02a], and [WN05] which followed this research direction. They started with a recognition system which was trained on a small amount of manually transcribed data and then decoded untranscribed audio data to obtain the automatically generated transcriptions for acoustic model training.

In [KW99], the impact of untranscribed data was examined on a recognizer that was trained with only a small amount of manually transcribed data. The authors investigated how many manually transcribed data were required to achieve reasonable results, and how good the quality of the automatically generated transcriptions was compared to the quality of manual transcriptions. The study concluded that each system can improve itself using automatic transcriptions. Furthermore, it was observed that in some cases a lot of data was necessary and the improvement was very slow. However, with increasing system performance, the self-learning process also accelerates. In [KW99], Kemp and Waibel also applied unsupervised training in combination with a confidence score to select accurate data for German ASR. In their experiments, the WER was improved from 32% to 21.4%. Furthermore, they conducted "oracle experiments", by simulating confidence measures with 100% correctness and showed that the WER of their system cannot be improved significantly beyond 21.4%. Lamel, Gauvain and Adda explored the concept of lightly supervised and unsupervised training with an iterative method in [LGA02a, LGA02b]. Their iterative refinement of transcriptions was based on several iterations of repeated Viterbi alignment of the generated transcriptions with the audio signal. The alignment was corrected manually and, afterwards, a standard EM-training was executed. Consecutive alignment and correction was repeated several times with an increasing amount of audio data and transcriptions. The authors also explored the use of closed captions which are partial transcriptions that depict the topic of the current speech segment. However, their results showed that the use of closed captions is difficult because of the missing distinction of speech and non-speech events, different word choices (synonyms), or alternating word order. For the task of unsupervised training, they tried to reduce the initial amount of data that has to be transcribed. With ten minutes of transcribed data and five iterations of unsupervised training, they almost reached the WER of the same recognizer trained with one hour of transcribed data. They also observed that closed caption filtering is not necessary for this method of iterative unsupervised training. Wessel and Ney applied unsupervised acoustic model training on broadcast news data [WN05]. They started with one hour up to five hours of manually transcribed data. They found that the more data they use, the better the recognition performance gets. However, the improvement is rather small and with one hour of manually transcribed data, they already got sufficiently good results.

4.2.2 Confidence score

The results of the above mentioned previous studies show that the use of confidence scores improved the performance of the unsupervised training approach. Hui Jiang conducted a survey on confidence measures in 2005 [Jia05]. Three

kinds of confidence measures are described: Predictor features, posterior probability, and utterance verification. Predictor features, for example n-best lists, acoustic stability, or hypothesis density, serve to distinguish false from correct results. However, none of these features is ideal and even a combination of several features does not lead to better performance. Posterior probability features try to estimate $p(X)$ from the fundamental equation of speech recognition. Examples for posterior probability features are filler based methods or lattice based confidences. The third group of confidence measures, utterance verification, formulates the problem of confidence measures as statistical hypothesis testing problem. Hypothesis 0, meaning that X was classified correctly, is compared to hypothesis 1, meaning that X was classified falsely, with a distance measure, for example likelihood ratio or Bayes factor. Hui Jiang concluded that lattice based confidence measures seem to provide good results, and have the advantage of incorporating language model scores. However, a general problem with confidences is that segmentation errors of the ASR system are not detected, but lead to bad confidences. Kemp and Schaaf [KS97] compared the performance of several word lattice based confidences. They compared features like gamma, hypothesis density, or acoustic stability. The overall conclusion of this paper was that all the confidence measures - besides gamma - give approximately the same results. Gamma, on the contrary, was more effective than all the other features combined, and therefore, was the clear winner. It is notable that the recognizer that was used to generate the confidence scores was quite strong with 13.2% WER on their dataset.

4.2.3 Cross-language bootstrapping

Schultz and Waibel introduced cross language transfer in [SW01a] and evaluated its application to Swedish based on GlobalPhone data. The idea is to borrow an existing acoustic model of one language for another language. Their experiments revealed that for Swedish, the results are independent of the baseline performance of the source language, as well as mostly independent of the language family of the source language. In their work, only the crosslingual effect was explored.

In [LGN09], the authors built a Polish ASR system by using Spanish ASR in combination with unsupervised training. A Spanish system on European Parliament plenary sessions (EPPS) speech data with an initial WER of approximately 10% was ported to Polish with manual phone mapping. The initial Polish model was refined through iterative recognition and re-training of 130 hours of Polish European Parliament audio data starting at an initial WER of approximately 60%. Their results are convincing but limited since the source

and target language are related. Therefore, the initial WER is accurate enough to apply unsupervised training.

4.3 Cross-language modeling based on phone mapping

Based on the initial situation described in Section 4.1, the first step is to transfer the acoustic models from the source languages to the target language to obtain an initial acoustic model which can be used for unsupervised training. That means, the acoustic models of the source languages are borrowed and directly used as initial model of the target language. For this task, the "cross-language transfer" technique is applied. This section presents the main idea of this technique and also two different implementations. Both implementations apply the phone mapping approach based on IPA.

4.3.1 General idea and implementation

Cross language transfer refers to the technique of applying a system developed in one language to recognize another language without using any training data of the new language. [SW01a] presented two principle ways of achieving a phone mapping: manual mapping using the IPA scheme or a mapping that was automatically derived from data using a target language phone recognizer. In this thesis, we evaluated the scenario that we do not have audio training data with transcriptions for developing an ASR, so we cannot build a phone recognizer. Therefore, we decided to use a manual mapping although in [SW01a], a slightly better performance is presented using an automatically derived mapping.

In the original implementation of the technique in [SW01a], the authors modified the acoustic models of the source languages, i.e. for each acoustic model of the context-independent HMM-states of the source languages, the acoustic model of the corresponding HMM-state based on the manual phone mapping of the target language was selected.

In contrast to the original approach of cross-language transfer [SW01a], we did not modify the acoustic model of the source languages, but the pronunciation dictionary of the target language, i.e. we modeled Czech words with phones of the other source languages. These mapped dictionaries allow the use of the acoustics of the source languages in combination with the pronunciation

dictionary and language model of the target language to decode the untranscribed audio data and, therefore, to generate automatic transcriptions. Figure 4.2 shows the idea of our modified cross-language transfer with Polish as source language and Czech as target language.

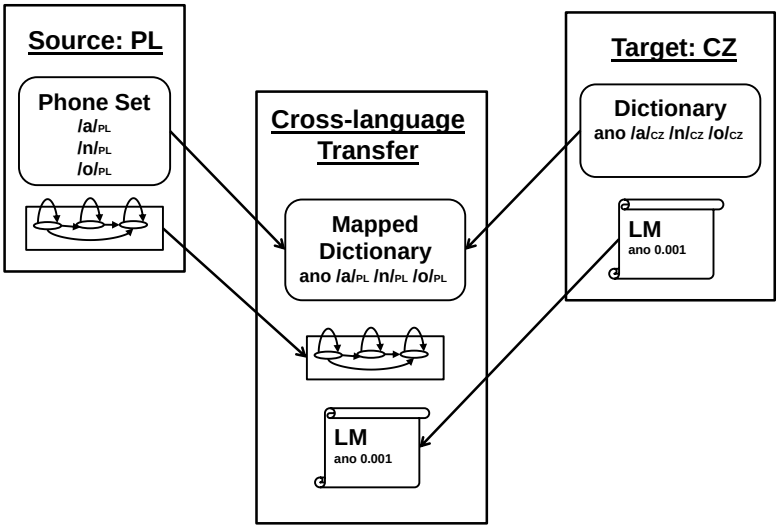


Figure 4.2: Modified cross-language transfer with Polish as source and Czech as target language

Consequently, in contrast to [SW01a], the modified approach will benefit from context similarities between languages by leveraging the context dependent acoustic models of the source languages.

4.3.2 Experiments and results

In these experiments, Czech serves as target language while two different language groups are used as source languages. The first language group contains closely related languages to Czech, such as Bulgarian (BL), Croatian (HR), Polish (PL), and Russian (RU). The languages English (EN), French (FR), German

(GE), and Spanish (SP) belong to the second group. Compared to the first group, the languages in the second group are not as related to Czech as the languages in the first group. The mappings for the evaluated languages are all created manually and are based on IPA similarities. If no phone with the same IPA symbol exists, a similar IPA phone is chosen based on articulatory features. Table 4.1 shows an overview of the mappings between Czech and the other source languages. Since Czech is the target language, each Czech phone needs a representative phone in each of the source languages. All the phones in table 4.1 are displayed in IPA notation in square brackets. The selected languages are quite different but still belong to the large Indo-European language family. Several consonant phones are equal in all nine languages and, therefore, not listed in Table 4.1. These phones are: [b], [d], [f], [g], [j], [k], [l], [m], [n], [p], [r], and [z].

We applied both the original and the modified cross-language transfer from the different source languages to Czech as target language. Table 4.2 compares the performance between the original and the modified cross-language transfer approach based on the Czech development set. It also shows the percentage of polyphone types from the target language covered by each source language, respectively. The results in table 4.2 indicate that the modified cross-language transfer outperforms the original approach for those source language that belong to the same language family as the target language. This is most likely due to the fact that words (and contexts) are more similar among the *Slavic* languages and, thus, better leverage the context dependent acoustic models after mapping the dictionary. However, we observed that the polyphone coverage and ASR performance are only loosely correlated, e.g. using the Polish acoustic model yields a better WER than the Bulgarian acoustic model although the Czech polyphones are better covered by the Bulgarian polyphones. Hence, we investigated the *Slavic* language family tree. Polish and Czech are both Western Slavic languages while Bulgarian is a Southern Slavic language, which can be a reason to explain the cross-language transfer results. In contrast, Schultz and Waibel [SW01a] did not observe any correlation between the ASR performance after applying cross-language transfer and the language similarity between source/target language. Linguistically closest to their target language Swedish is German, but Turkish and Korean worked best in their experiments.

4.4 Multilingual A-Stabil - A Multilingual Confidence Score

The basic idea of unsupervised training is to improve an acoustic model by iterative recognition of audio data without manual transcriptions. Instead, au-

Table 4.1: *Overview of phone mappings from the 8 source languages to Czech*

CZ	BL	EN	FR	GE	HR	PL	RU	SP
c [ʦ]	[ʦ]	[s]	[s]	[ʦ]	[ʦ]	[c]	[ʦ]	[s]
ch [ʧ]	[ʧ]	[ʧ]	[ʃ]	[x]	[ʧ]	[ʧ]	[ʧ]	[ʧ]
dj [ʄ]	[dʲ]	[θ]	[d]	[d]	[d]	[d]	[d]	[ð]
h [ɦ]	[k]	[h]	[h]	[h]	[x]	[ɦ]	[h]	[ɣ]
mg [ɱg]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]
nj [ɲ]	[nʲ]	[n]	[ɲ]	[n]	[nʲ]	[n]	[nʲ]	[ɲ]
ng [ŋ]	[n]	[ŋ]	[ŋ]	[ŋ]	[nʲ]	[n]	[n]	[ŋ]
rsh [r]	[r]	[ɹ]	[ʁ]	[ʁ]	[r]	[r]	[r]	[ɾ]
rzh [r]	[r]	[ɹ]	[ʁ]	[ʁ]	[r]	[r]	[r]	[ɾ]
sh [ʃ]	[ʃ]	[ʃ]	[ʃ]	[ʃ]	[sʲ]	[ʃ]	[ʃ]	[s]
tj [ʧ]	[ʧʲ]	[ʧ]	[ʧ]	[ʧ]	[ʧ]	[ʧ]	[ʧʲ]	[ʧ]
x [x]	[x]	[ʃ]	[ʁ]	[x]	[sʲ]	[sʲ]	[x]	[x]
zh [ʒ]	[ʒ]	[ʤ]	[ʒ]	[ʃ]	[zʲ]	[ʒ]	[ʐ]	[z]
a [ʌ]	[ə]	[a]	[a]	[ʌ]	[ʌ]	[ʌ]	[ʌ]	[a]
aa [ʌ]	[ʌ]	[ɒ]	[ɒ]	[a:]	[ʌ]	[ʌ]	[ʌ]	[a]
aw [au]	[ʌ]	[au]	[ɐ]	[aʊ]	[ʌ]	[ʌ]	[ʌ]	[au]
e [ɛ]	[ɛ]	[e]	[e]	[e]	[ɛ]	[ɛ]	[ɛ]	[e]
ee [ɛ:]	[ɛ]	[e]	[ɛ]	[e:]	[ɛ]	[ɛ]	[ɛ]	[e]
ew [iw]	[ɛ]	[eɪ]	[ø]	[ɔy]	[ɛ]	[ɛ]	[ɛ]	[eu]
i [ɪ]	[ɪ]	[ɪ]	[i]	[i]	[ɪ]	[ɪ]	[ɪ]	[i]
ii [i]	[i]	[i]	[i]	[i:]	[i]	[i]	[i]	[i]
o [o]	[o]	[ɔ]	[o]	[o]	[o]	[o]	[o]	[o]
oo [o]	[o]	[ɔ]	[ɔ]	[o:]	[o]	[o]	[o]	[o]
ow [ou]	[o]	[oʊ]	[o]	[o]	[o]	[o]	[o]	[o]
u [ʊ]	[u]	[ʊ]	[u]	[u]	[u]	[u]	[u]	[u]
uu [u]	[u]	[u]	[u]	[u:]	[u]	[u]	[u]	[u]

tomatically generated transcriptions are used to re-train or adapt the acoustic model. For an effective use of available acoustic data, it is important to utilize confidence measures to select or weight the contributions of the audio data so that only training data with accurate automatic transcriptions are used. In this section, we describe the investigation of confidence scores and propose a new method called "Multilingual A-stabil" which is based on ASR for multiple languages. We show that "Multilingual A-stabil" suits better than other confidence score measures when the acoustic model is poorly estimated.

Table 4.2: *Original vs modified cross-language transfer (WER)*

Languages	Original	Modified	abs. Δ	Polyphone Coverage
Bulgarian (BG)	67.0%	61.0%	6%	16.9%
Croatian (HR)	68.0%	57.2%	10.8%	15.6%
Polish (PL)	67.7%	55.8%	11.9%	13.2%
Russian (RU)	72.5%	64.3%	8.2%	10.0%
Spanish (SP)	85.4%	87.2%	-1.8%	6.8%
German (GE)	75.2%	75.2%	0%	6.4%
French (FR)	84.5%	95.2%	-10.7%	2.0%
English (EN)	87.4%	99.8%	-12.6%	0.4%

4.4.1 Investigation of confidence scores

In [KS97], "gamma" and "A-stabil" were presented and have been widely applied to unsupervised training afterwards. The authors showed a high correlation between these confidence scores and the word error rate of the speech recognition system. However in their experiments, a strong German ASR with high accuracy on the test set was used. In our experiments, when the initial acoustic model is obtained by using the crosslingual transfer technique, the ASR system is rather weak. Therefore, we regarded the robustness of gamma and A-stabil to figure out whether they are a suitable confidence score in our task.

To evaluate gamma and A-stabil, we plot the performance (WER) over selected confidence thresholds. We used the CZ system to decode the development set and evaluated the WER of all the words occurring in the specified confidence interval using steps of 0.1. Figure 4.3 compares gamma and A-stabil for two systems: a CZ system trained on about 23 hours of CZ training data and a CZ system resulting from cross-language transfer. The WER is 22.7% and 55.8% on the Czech test set, respectively. During decoding, the language model weight and the insertion penalty was set to 26 and 0. To compute A-stabil, we generated 100 alternative hypotheses by varying the language weight from 35 to 44 with a step size of 1 and the insertion penalty from -8 to 10 with a step size of 2. The figure shows that gamma and A-stabil work very well with well-trained acoustic models, but have problems with the initial acoustic models generated by the cross-language transfer. Due to the poor performance of these confidence scores, it is difficult to apply unsupervised acoustic model training. Hence, a more robust confidence score is required.

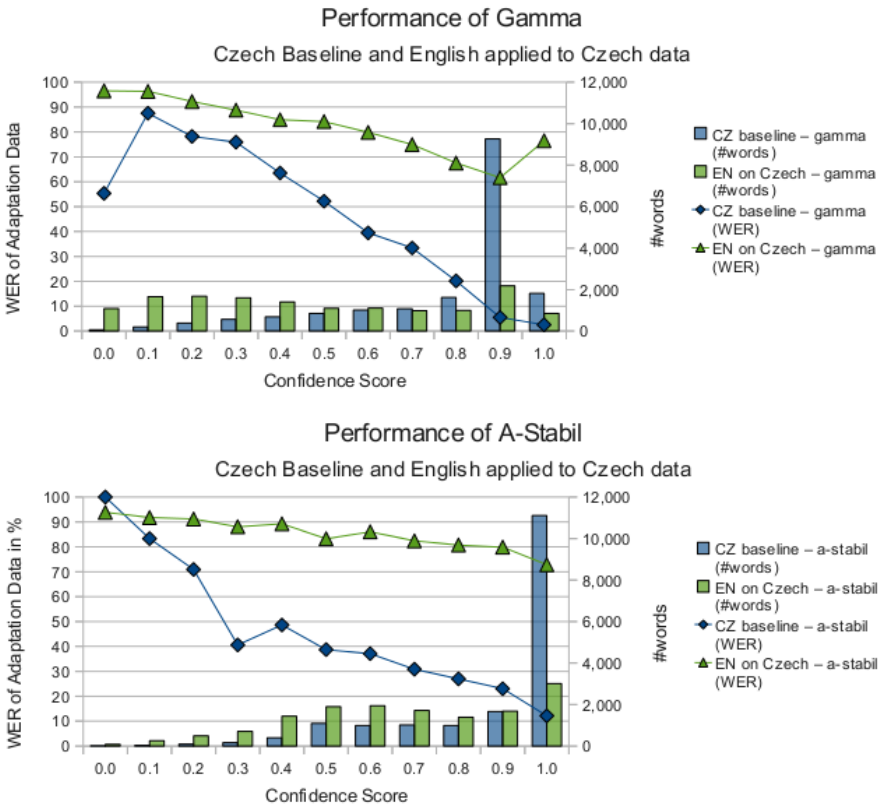


Figure 4.3: The plot of recognition errors over gamma (and A-stabil) using a well-trained Czech acoustic model and an initial cross-language acoustic model (Polish) [Kra11]

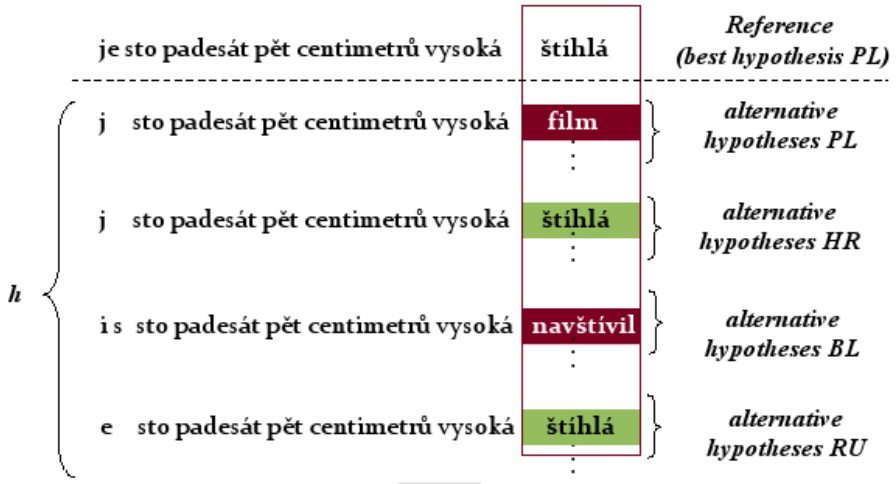


Figure 4.4: "Multilingual A-stabil" method to compute word-based confidence scores

4.4.2 Multilingual A-Stabil

Based on the idea of A-stabil, we propose a new method to compute confidence scores using acoustic models from n different languages. First, for the acoustic model of each language the word lattices are extracted. To generate the alternative hypotheses, we vary the weight of the language model and the word insert penalty of each language. Instead of using only alternative hypotheses of one language, we merge all sets of alternative hypotheses from different acoustic models to obtain a multilingual arbiter. Afterwards, the frequency of each word of the reference output is computed based on this set normalized by the number of alternative hypotheses. By applying this technique, the multilingual arbiter uses divers information from different languages which might be helpful to compensate mismatching phone sets between languages. Moreover, the multilingual arbiter does not force the system to merge the acoustic units, such as phones or subphones across languages. In contrast, it collects all the information provided by each monolingual speech recognizer and lets the system choose which information from which language should be used by counting the frequency of the word hypothesis. Figure 4.4 illustrates the new method to compute word-based confidence scores. In this example, the Czech acoustic model was generated by using the cross-language transfer technique with Polish as source language. We used this model to decode the audio data

and obtained the best hypothesis, which is referred to as reference in Figure 4.4. Afterwards, we computed the confidence score for each word in this reference. For this, we used not only the acoustic model from Polish but also from Croatian, Bulgarian and Russian to generate alternative hypotheses. Finally, the reference words are counted in these alternative hypotheses with consideration of the correct time steps. The following equation shows how to compute the "multilingual A-stabil" confidence score from these counts:

$$\text{multilingual A-stabil} = \frac{\#occurrence(\text{reference word})}{h} \tag{4.1}$$

where h is the total number of the alternative hypotheses.

Note that the original definition of A-stabil is a specialization of the new method with $n = 1$, that means monolingual. Hence, we refer to it as "multilingual A-stabil". Figure 4.5 shows the relation of the recognition error and this score. We detect a very high correlation between the multilingual A-stabil and the recognition error for both well-trained acoustic models and poorly estimated acoustic models. In contrast to gamma and A-stabil, multilingual A-stabil is much more robust against poor ASR performance. Furthermore, the quality of the confidence score increases significantly if four languages are used. That means, the WER is comparatively low for high confidence scores considering the high overall WERs of all four recognizers. Here again, the x-axis represents confidence score intervals, the left y-axis the WER for all the words in the current confidence score interval, and the right y-axis the number of words in the current confidence score interval. The plots of WER over confidence score (yellow and red curve in Figure 4.5) show a much higher correlation than the plots of gamma and original A-stabil using the initial acoustic model (green curve in Figure 4.3). The bars in Figure 4.5 represent the amount of data within a confidence score interval. Since the WER of the initial acoustic model is quite high, there is only a small amount of data with high confidence scores. However, with further adaptations of the initial recognizers and thus, rising recognition accuracy, more words with high confidence scores can be obtained. Throughout the adaptation and data selection process, sufficient data quality is ensured with selection of an adequate threshold. The correlation between confidence score and WER can be observed for both language groups, Slavic languages and resource rich languages, similarly. That means "multilingual A-stabil" seems to be reliable even for languages that are not closely related to the target language. In this case, EN, FR, GE, SP are from the same language family (Indo-European) but not as close to CZ as BG, HR, PL, and RU.

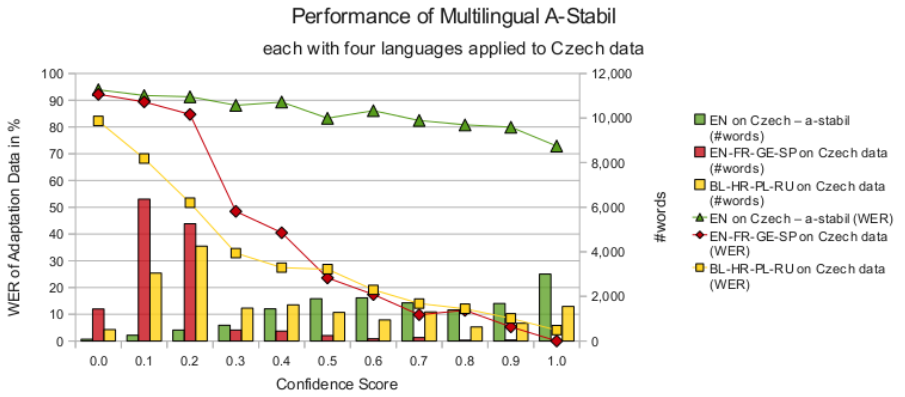


Figure 4.5: Performance of multilingual A-stabil confidence scores calculated with four languages (EN, FR, GE, SP and BL, HR, PL, RU) compared to the performance of A-stabil for one language (EN) [Kra11]

4.4.3 Threshold selection

Not only the confidence score itself but also the chosen threshold is crucial for the quality of the data that will be selected. After calculating the confidence scores, every word that has a score above the selected threshold will be selected as adaptation data. Choosing a threshold too low will lead to a greater amount of adaptation data with less quality. A threshold too high will not select enough data though ensures a very high quality. The optimal threshold is a trade-off between data quality and amount of data. That means, data with a sufficiently high quality should be selected. Since finding the optimal threshold is complex and its verification needs a lot of computation time, we propose an approach to heuristically obtain a reasonable threshold based on observations of the quality of automatic transcriptions of the development set. Figure 4.6 shows the performance of "multilingual A-stabil" for different numbers of languages. On the x-axis, the confidence score intervals are listed, meaning scores from 0 to 0.1 for the first points, from 0.1 to 0.2 for the second ones and so on. The y-axis shows the word error rate of the adaptation data for the corresponding confidence score interval. The curves for two and four languages clearly lie below the curve for one language (A-stabil) and, therefore, provide a superior confidence measure. The drop in WER at 0.2 (for four languages) and 0.5 (for two languages) indicate the multilingual effect, because at theses points more than one language has to agree to the same hypothesis word to reach the targeted confidence score. If we want to select a reasonable threshold, a first thought is

to use the effect of multilingualism and select as many data as possible. Obviously, "multilingual A-stabil" for only one language (green line in Figure 4.6) is equal to the original A-stabil and, therefore, does not provide sufficient data quality for any given threshold. As soon as at least two languages vote for the multilingual confidence score, the WER of the adaptation data drops substantially. Thus for N languages, a threshold of $1/N + \mathit{offset}$ should give reasonable results. The *offset* can be chosen in such a manner that a word has to occur a certain amount of times in all considered languages. For example, the minimal threshold for four languages would be $1/4 = 0.25$. That means each word with a confidence score greater than 0.25 has to occur not only in alternative hypotheses of one language, but at least once in an alternative hypothesis of another language. This heuristic works well for $N = 2$ and $N = 4$ languages as shown in Figure 4.6. However, it does not guarantee the best choice of threshold.

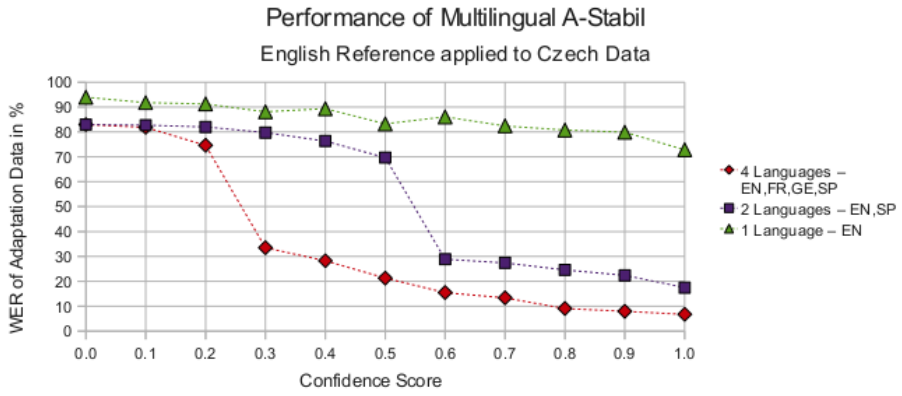


Figure 4.6: Performance of multilingual A-stabil for different numbers of languages - one, two, and four languages [Kra11]

4.5 Multilingual unsupervised training framework

In this section, we present our multilingual unsupervised training framework which combines cross-language transfer technique and unsupervised training with the help of the "multilingual A-stabil" confidence score to build an ASR system without any transcribed data. As mentioned in Section 4.1, we assume to have ASR systems for several source languages, as well as a language model,

s pronunciation dictionary and untranscribed data of the target language. The main idea of the training framework is to select audio data and automatically generate transcriptions which can be used to adapt or train the acoustic model of the target language based on "multilingual A-stabil". At the beginning, since there is no initial acoustic model of the target language, the cross language transfer technique is applied for each source language. In this step, several phone mappings between the source and the target languages have to be created manually. Afterwards, several decoding processes are run in parallel with all the initial acoustic models to transcribe the audio data automatically. Based on the decoding results, alternative hypotheses are created and collected from all the source languages. The resulting pool of hypotheses is then used to compute the "multilingual A-stabil" score for the hypotheses of each source language individually. Since the initial acoustic model is quite weak at the beginning of the process, we apply acoustic model adaptation to improve the performance of the recognizer until a sufficient amount of training data could be selected. The adaptation itself is a common MAP adaptation. The adaptation data of each iteration are selected from the current recognition results (of this iteration). It is therefore only used for the current iteration, that means two iterations only cohere in using the same initial - or adapted - recognizer.

This process applied in this research is independent of source or target languages. The same iterative recognizer adaptation is applied for each source language. That means if "multilingual A-stabil" is computed from more than one language, we have several adaptation processes for the same target language in parallel. The multilingual framework sets up a generic structure for the parallel adaptation processes. Figure 4.7 shows an overview of the framework. Each source language recognizer is bootstrapped to the target language and afterwards adapted separately.

For each target language, several source language folders are created, with each of them containing the whole process structure. Additionally, the main folder (target language) contains representations of the language model, the pronunciation dictionary, and the audio database for the target language. These components are global and similarly used for all source languages. Each source language folder mainly consists of two parts:

- The initial source language recognizer or a target language recognizer created via bootstrapping and acoustic model training from selected adaptation data.

- The decoding/adaptation cycle, in which the recognizer is iteratively improved. The framework structure is explained in more detail in the next paragraph.

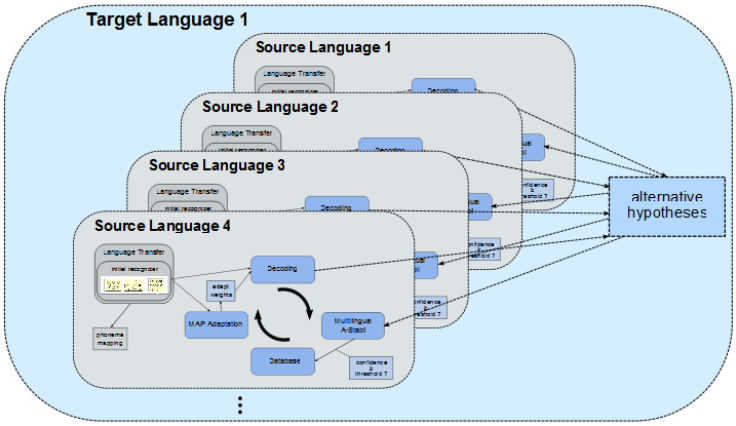


Figure 4.7: Overview of the multilingual unsupervised training framework [Kra11]

The source language folder contents are depicted in Figure 4.8. They embody the core of the multilingual framework. Each source language iteratively follows the adaptation process that is illustrated on the right side of Figure 4.8. Directories for bootstrapping a new recognizer and for the decoding/adaptation cycle are situated in each source language folder. The decoding/adaptation process is as follows: The bootstrapped source language recognizer is used to decode the audio data. Then, "multilingual A-stabil" is computed from the pool of available alternative hypotheses. Afterwards, an adaptation database is constructed and used to adapt the bootstrapped source recognizer. This process is performed iteratively until the amount of collected adaptation data is not sufficient anymore.

4.6 Experiments and results

4.6.1 Experimental setup

For this research, we selected French, German, Spanish, Bulgarian, Polish, Croatian, Russian, Czech, and Vietnamese from the GlobalPhone corpus. In addition, we used English speech data from WSJ0. Czech and Vietnamese serve as target languages and the remaining ones are used as source languages. We split the source languages into two different categories in order to perform the

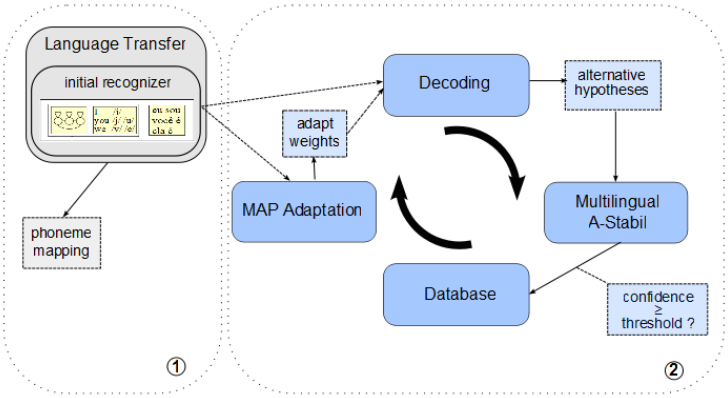


Figure 4.8: Multilingual unsupervised training framework with boostrapping/initial recognizer (1) and adaptation circle (2) [Kra11]

experiments: The first one called *Big4* contains European, resource-rich languages, namely English, French, German, and Spanish. The second one consists of four different *Slavic* languages, namely Bulgarian, Croatian, Polish and Russian. The idea is to increase the difficulty of the experiments step by step to explore the generalization ability of the framework. According to this, the experiments are categorized into three levels as follows:

- Level I: Using *Slavic* languages to bootstrap Czech ASR - source languages and target language are closely related since they all belong to the *Slavic* language family.
- Level II: Using *Big4* languages to bootstrap Czech ASR - source languages and target language stem from the *Indo-European* language family, but are not as close related as in the level I.
- Level III: Using *Big4* and *Slavic* languages to bootstrap Vietnamese ASR - source languages and target language are not related since the source languages are *Indo-European* languages and the target language is a *Sino-Tibetan* language.

4.6.2 Closely related languages vs resource-rich languages

The first two experiments were conducted by using Czech as the target language and two different groups of source languages: *Slavic* and *Big4* languages.

The motivation is to look at the final ASR performance while slightly decreasing the similarity between source languages and the target language. Czech and the *Slavic* group belong to the Slavic language family while the languages of *Big4* stem from Germanic and Romance language families. However, all of them belong to the Indo-European language family. Furthermore, the data of the resource rich languages English, French, German and Spanish are easier to obtain than the data of the four Slavic languages. Therefore, it is more likely that we have ASR systems of those resource rich languages to bootstrap the ASR system for a new language.

Iterative generation of automatic transcriptions

In the case of Russian, Bulgarian, Croatian and Polish, we applied the modified cross-language transfer without re-training to generate the initial acoustic models. The word error rate is around 60% on the Czech development set. In contrast, we used the original cross-language transfer for English (EN), French (FR), German (GE), and Spanish (SP). The WER is relatively high, with 87.35% for EN, 84.52% for FR, 75.30% for GE, and 85.42%. With these initial models, we recognized the Czech training data and selected appropriate adaptation data using "multilingual A-stabil" confidence scores. Based on the heuristic described in 4.4.3, we chose 0.3 as the threshold to select the training data. Therefore, words have to occur in alternative hypotheses from more than one language in order to be selected. Table 4.3 shows the amount of selected data after each iteration in percentage of all the untranscribed data and their quality in terms of WER. The results show that using Slavic languages, we could select more training data (28% relative) with more accurate automatic transcriptions (31.6% relative) compared to using resource-rich languages. For both cases, we observed that after four iterations the amount of selected data increased rather slightly. In the case of resource-rich languages, the quality of transcriptions even got slightly worse. Therefore, we stopped the adaptation circle after four iterations.

Cross-language boostrapping

After acoustic training data with high quality transcriptions have been selected, we used the bootstrapping approach to train the Czech ASR by using the multilingual acoustic model inventory which was trained earlier from seven GlobalPhone languages [SW01b]. To bootstrap the system, an initial state alignment was produced by selecting the closest matching acoustic models from the multilingual inventory as seeds. The closest match was derived from an IPA-based phone mapping. After initialization, the system was completely rebuilt using

Table 4.3: *Iteratively enlarging the amount of training data with automatic transcriptions: results for the source languages Polish and German*

Iteration	Amount of data	% of all data	Quality (WER in %)
Polish:			
1	5.5h	23.9	25.0%
2	14.3h	62.2	17.0%
3	15.9h	69.1	16.5%
4	16.4h	71.0	16.0%
German:			
1	2.3h	10.1	27.1%
2	8.7h	37.8	22.9%
3	10.1h	43.6	23.4%
4	10.2h	44.2	23.5%

the selected data. We trained a quintphone system with 1,500 contexts by applying merge&split and Viterbi training. Figure 4.9 shows the performance of the four different systems which were trained with four different selected data sets on the Czech development set. By using the Slavic languages as source, the WER ranges from 23.0% to 23.6%. By comparison, the average WER is about 26.6% if the resource rich languages served as source languages. Obviously, using related languages as source, we can obtain a better initial acoustic model and, therefore, more training data with more accurate automatic transcriptions to train a Czech ASR system. The best WER was achieved using the acoustic training data which was generated by modified cross-language transfer using Russian as source language.

To increase the amount of the acoustic training data, we decoded the training data again using the acoustic model from the previous iteration and selected data with high confidence of "multilingual A-stabil". In the case of using Slavic languages and resource-rich languages, we obtained about 18.5h (80%) and 16.8h (73%) of the training data with automatic transcriptions which have 14.5% and 14.6% WER respectively. For the second iteration, we used the acoustic model from the first iteration to generate the state alignment and trained the system with the same parameters as in iteration 1 afterwards. Since more training data was selected, we increased the number of contexts to 2,000. The best system generated by Slavic languages has 22.7% WER on the development set and 22.3% WER on the evaluation set. In contrast, we obtained 23.3% WER on the development set and 22.8% WER on the evaluation set by using the resource-rich languages. The results indicate that there is only a minor difference in terms of WER between using related and non-related source languages.

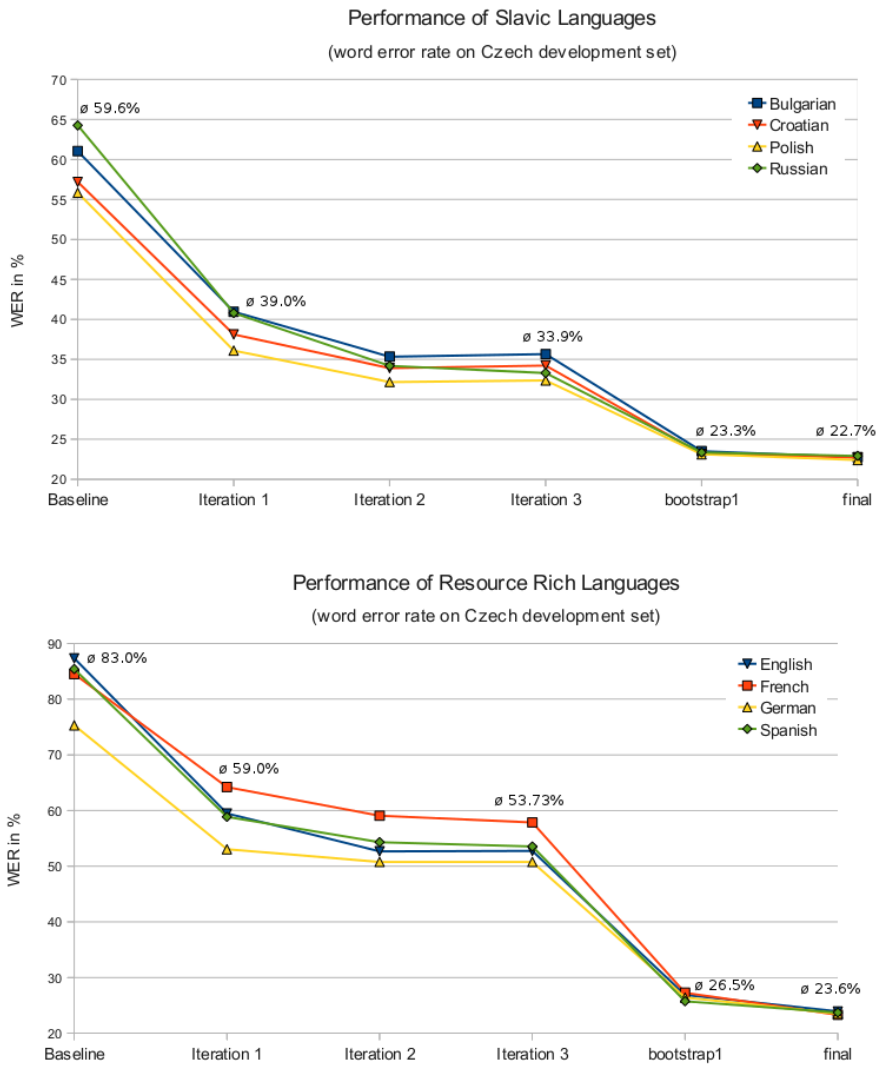


Figure 4.9: Development of speech recognizer quality measured in WER on the Czech development set using the Slavic source languages vs. resource rich languages [Kra11]

Furthermore, the final Czech ASR built with our proposed training framework has almost the same performance as the Czech baseline system trained with manual transcriptions. The WER of the baseline system is 22.3% on the evaluation set.

4.6.3 Under-resourced languages - a study for Vietnamese

In the third experiment, we built an ASR system for an under-resourced language - Vietnamese in this case - using the multilingual training framework with up to six different Indo-Eurpoean languages as source languages. With this experiment, we simulate one of the most challenging cases in which the source languages and the target language are not related since Vietnamese belongs to the Sino-Tibetan language family. Furthermore, we use different numbers of source languages and regard the impact of the number of the source languages on the final Vietnamese ASR performance.

Syllable- vs. Word-based

In order to improve "Multilingual A-stabil" for the case of Vietnamese, we compute the confidence score on the syllable level. This means, we split Vietnamese words into syllables before computing the confidence score. We found the voting process to be more efficient at syllable level than at word level. Therefore, we can extract more data using the same confidence threshold. Another benefit of generating automatic transcriptions on syllable level is that co-articulation effects can be modeled by an adaptation or training process. Table 4.4 shows the amount of data and the quality of automatic transcriptions in terms of SyllER by applying "Multilingual A-stabil" at syllable and word level for four different languages (EN, SP, GE and FR) with a threshold of 0.3 for the first iteration. It indicates that we gain 24% more training data by applying "Multilingual A-stabil" at syllable level while achieving almost the same transcription quality. Therefore, we applied "Multilingual A-stabil" at syllable level for the remaining experiments.

Table 4.4: *Syllable- vs. Word-based "Multilingual A-stabil"*

	Amount	SyllER	Rel. Gain
Word-based	0.75h	51.54%	
Syllable-based	0.93h	52.83%	+24%

Iterative automatic generation of transcriptions

We started by applying cross-language transfer based on English (EN), French (FR), German (GE), Spanish (SP), Bulgarian (BG) and Polish (PL) acoustic models without any re-training in order to recognize the Vietnamese development set. The SyllER was very high with 90.93% for EN, 92.81% for FR, 93.49% for GE, 89.72% for SP, for 88.49% BG and 86.58% for PL which indicates the challenges of building a Vietnamese ASR system from scratch without any transcriptions. With these initial models, we decoded the Vietnamese training data and selected appropriate adaptation data using the "multilingual A-stabil" confidence scores. As we observed in 4.4.3, the SyllER drops rapidly when we select those syllables which are voted for by at least two languages. To reflect this with two, four, and six languages, 0.6, 0.3, and 0.2 were chosen as confidence score thresholds respectively. We terminated the process after four iterations, since the gains of the amount of selected data and the quality of the automatic transcriptions seem to saturate. Figure 4.10 displays the amount of selected data over the iterations in percentage of the number of all the untranscribed syllables. The figure also shows the resulting transcription quality in terms of SyllER by using two (EN, SP), four (EN, SP, FR and GE) and six source languages (EN, SP, FR, GE, BG and PL) that cover 26, 27, and 28 of the 39 Vietnamese phones. The results indicate a close relation between the amount of extracted data and the number of languages respectively the phone coverage. The more target languages we use in our training framework, the more phones we can cover from the target language and, thereby, the more data we are able to select. However, Figure 4.10 also indicates that the quality of the automatic transcriptions gets slightly worse if we use more source languages.

Cross-language bootstrapping

We used the selected Vietnamese acoustic training data with the automatic transcriptions from the initial step to train the Vietnamese acoustic model in this final step. First, we trained the multilingual inventory with all the existing data from the source languages by applying an IPA-based phone merging [Ass99]. The closest match is derived manually according to IPA similarity. Table 4.5 summarizes the performance of multilingual acoustic models MM2 (EN, SP), MM4 (EN, FR, GE, and SP), and MM6 (EN, SP, FR, GE, BG, and PL) after cross-language transfer on the development set. The results indicate that a larger number of source languages used for the training of the multilingual acoustic models improves the cross-language transfer performance on the Vietnamese development set. Therefore, the quality of state alignment might be improved. Afterwards, an initial state alignment for the Vietnamese training data is produced by determining the closest matching acoustic models from the multi-

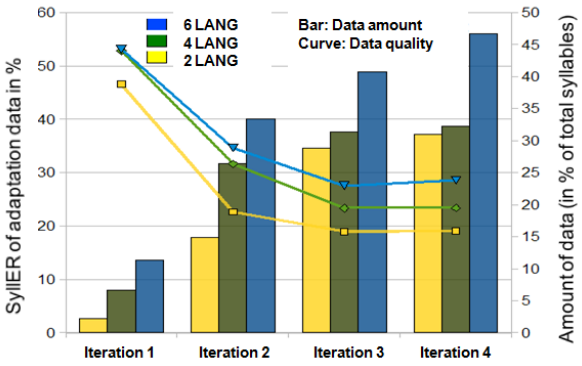


Figure 4.10: Amount of selected data given in percentage of all syllables and the corresponding resulting transcription quality in terms of SyllER

Table 4.5: *Cross-language transfer performance (on VN dev set) of multilingual acoustic model MM2 (EN, SP), MM4 (EN, SP, FR and GE) and MM6 (EN, SP, FR, GE, BG and PL)*

Systems	SyllER	Rel. Delta
MM2	87.54%	
MM4	82.35%	+5.9%
MM6	76.45%	+7.2%

lingual inventory as seeds. Then, the Vietnamese system is completely rebuilt using the seed acoustic models and the selected data for training (one data set per source language). We built a quintphone system with 1,500 contexts with the same training procedure described in paragraph 4.6.2.

To increase the amount of selected acoustic training data, we again decoded the training data. For the second iteration, we used the acoustic model from the first iteration to generate the state alignments and then trained the system with 2,000 quintphone contexts. Figure 4.11 summarizes the performance of our Vietnamese ASR system after the second iteration in terms of SyllER by using two (EN, SP), four (EN, SP, FR and GE), and six source languages (EN, SP, FR, GE, BG and PL). The resulting best system achieves 16.8% SyllER on the Vietnamese development set and 16.1% SyllER on the evaluation set. The results show that iterative unsupervised training with "multilingual A-Stabil" results

in accurate automatic transcriptions. They allow to further improve the acoustic model of the target language. Compared to the baseline system, trained on about 22 hours of transcribed data which achieves a SyllER of 14.3%, the final results are quite close. However, they are still worse than our best system where we applied various language specific optimization steps and achieved 11.8% SyllER [VS09].

Furthermore, using more different languages for our multilingual unsupervised training framework results in better performance of the final Vietnamese ASR system. However, the difference between using four and six source languages was minor, while the training time increased dramatically. Every time, when one more source language was used, we needed to decode all the training data five times in our experiments. When the training data is large, it might not be worthwhile to increase the number of source languages, since the difference in SyllER is minor. In all the experiments, four source languages seemed to be enough to successfully build an ASR systems for new languages without any transcribed data with our multilingual unsupervised training framework.

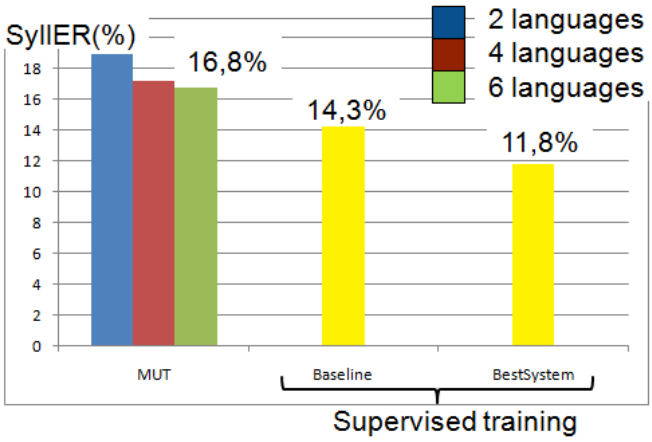


Figure 4.11: Cross-language bootstrapping for Vietnamese by using two (EN, SP), four (EN, SP, GE, FR) and all six languages

4.7 Summary

In this chapter, we presented the multilingual unsupervised training approach to rapidly build an ASR system for new languages without any transcribed

data. We explored different implementations of cross language transfer techniques and its performance on related and non-related language pairs. Furthermore, we proposed a new method to compute word-based confidence scores called "Multilingual A-stabil" which works well not only with well trained acoustic models but also with a poorly estimated acoustic model. Finally, we described the whole framework that uses several ASR systems from different languages and the available resources of the target language, such as untranscribed audio data, text data and pronunciation dictionary to build an ASR system for the target language.

To evaluate the framework, we conducted three experiments with increasing level of difficulty. The experimental results indicate that our proposed framework can be applied to build an ASR system without any transcribed data for new languages. We were able to build ASR systems for new languages even if the source languages and target language were not related in terms of language family and also polyphone coverage. However, using related source languages led to a better ASR system of the target language. In our experiments, we obtained 5% relative improvement by using Slavic source languages to bootstrap Czech ASR instead of using *Big4* source languages.

In the first two experiments, the source languages were varied from Slavic languages to Big4 languages which increased the word error rate of the cross-language transfer system by up to 20% relative on the Czech development set. This resulted in a gap of about 5% relative between the final Czech ASR systems. The relation between the performance of the cross-language transfer system and the final ASR system is obvious. That also means, the accuracy of the phone mapping between the source languages and the target language which might change the performance of the cross-language transfer system a bit should not have any significant impact on the final ASR system of the target language.

Our framework demonstrated its success on different experimental setups and proved to be useful to build ASR systems without any transcribed data. Therefore, it will save a lot of time and cost by developing ASR systems for new languages. Moreover, to our knowledge it is the first time in literature to show that it is possible to bootstrap an LVCSR system for a language which is not related to the source languages without any transcribed data. The limitation of the framework is the need of a manual phone mapping between source and target language, as well as the pronunciation dictionary and language model of the target language. That means, first, if the language model is not available or not strong enough due to insufficient text data or second, more extremely, if the language does not have e.g. a written system or any knowledge about the phone inventory, the framework may not be usable. In these cases, it might be worthwhile to transcribe the data manually, or to assume that prompts are

sufficiently close to the transcriptions using toolkits, such as RLAT [RLA12] or Woefzela [DVBD+11].

CHAPTER 5

Multilingual Bottle-Neck Features and Their Application To New Languages

Using Bottle-Neck features is one way to integrate neural networks into ASR systems at feature level. Previous works showed their success improving state-of-the-art ASR performance on different tasks and datasets. This chapter explores the use of multilingual data to improve Bottle-Neck features for ASR for new languages. The study starts with our proposal of an initialization scheme using multilingual MLP. Afterwards, the impact of the amount of data and languages as well as the similarity of source and target languages on the final ASR performance are investigated. The chapter ends with a detailed analysis of the output of the Bottle-Neck hidden layer to provide a better understanding of the behavior of those features in the context of multilingual and crosslingual characteristics.

5.1 Introduction

Cepstral features have been widely used in many speech processing applications for many years and have become standard features. At the beginning of 2000, Hermansky proposed Tandem features [HDS00] which allow the integration of neural network techniques to extract features for a speech recognition system. The idea is to use the posterior of a neural network as features. Afterwards in 2007, Bottle-Neck features were proposed by Grezl [GKKC07]. Instead of using the values of the output layer of a neural network (Tandem features), he used the output of the hidden layer (Bottle-Neck features) which is supposed to store the most important information of the input features e.g. cepstral features. They are known as multilayer perceptron (MLP) features in the literature. In many setups and experimental results, MLP features proved to be a high discriminative power and very robust against speaker and environmental variations. Furthermore, a very important characteristic of those features, which is related to this thesis, is the possibility to use multilingual data to make them robust against language variation, and therefore improve the final ASR performance. There are several interesting crosslingual and multilingual studies which showed that MLP features are language independent (summarized in 5.2), i.e. an MLP can be trained with data of one language or multiple languages and then used to extract features for a new language.

In this thesis, we focus on using Bottle-Neck features to train the ASR system. However, to extract the Bottle-Neck features, an accurate MLP has to be trained first. The machine learning research community showed that an MLP highly depends on its initialization and has a lot of parameters. At this point, the thesis presents an innovative approach to first train a multilingual MLP with a large amount of multilingual data and, then to use it to initialize the MLP training process for new languages. The goal is to achieve a robust initialization scheme and to allow training an MLP with many parameters using only a small amount of training data. For that, we propose a method to train a multilingual MLP which covers not only the multilingual phones but also the phones of the target language. That means, the final Bottle-Neck features are extracted from an MLP which has learned the multilingual data and the data of the target language. Therefore, we refer to them as *multilingual Bottle-Neck features*. To have a better understanding about the initialization scheme, we explored the impact of the number of languages as well as the similarity of the source and target language and the final ASR performance. Finally, a visualization of the output of the Bottle-Neck hidden layer is performed using t-Distributed Stochastic Neighbor Embedding [VdMH08].

5.2 Related work

This section provides a short summary of researches related to MLP features in a multilingual and crosslingual context. Many of them demonstrate that MLP features are language independent. In many papers, it was shown that features extracted from an MLP which was trained with one language can be used for another language.

For example, the authors of [TGH06] showed that features extracted from an English-trained MLP improved Mandarin and Arabic ASR performance over the spectral feature (MFCC) baseline system. Crosslingual portability of MLP features from English to Hungarian was investigated by using English-trained phones and articulatory feature MLPs for a Hungarian ASR system in [TFGK08]. Furthermore, a crosslingual MLP adaptation approach was investigated, in which the input-to-hidden weights and the hidden biases of the MLP corresponding to the Hungarian language were initialized by English-trained MLP weights, while the hidden-to-output weights and output biases were initialized randomly. The results indicated that crosslingual adaptation often outperforms cases, in which the MLP features are extracted from a monolingual MLP.

In [CMDL$^+$07], it was explored how portable phone and articulatory feature based tandem features are in a different language without re-training. Their results showed that articulatory feature based tandem features are comparable to the phone-based ones if the MLPs are trained and tested on the same language. However, the phone based approach is significantly better on a new language without re-training.

Imseng et al. [IBD10] investigated multilingual MLP features on five European languages, namely English, Italian, Spanish, Swiss French, and Swiss German from the Speech-Dat(II) corpus. They trained a multilingual MLP to classify context-independent phones and integrated it directly into the preprocessing step for monolingual ASR. Their studies indicate that shared multilingual MLP feature extraction yields the best results.

Plahl et al. [PSN11] trained several NNs with a hierarchical structure with and without Bottle-Neck topology. They showed that the topology of the NN is more important than the training language, since almost all the NN features achieve similar results, irrespective of whether training and testing languages match. They obtained the best results on French and German by using the (crosslingual) NN which has been trained on Chinese or English data without adaptation.

In [TGH10, TGH12], Thomas et al. demonstrated how to use data from multiple languages to extract features for an under-resourced language and, therefore, improve the ASR performance. They proposed to use a data-driven approach

in which no knowledge about the phone set of the target languages was needed. In [VKG+12], the language independent character of Bottle-Neck features was demonstrated on the GlobalPhone database. Improvements were observed by using multilingual Bottle-Neck features.

5.3 Multilingual multilayer perceptron and its application to new languages

5.3.1 Multilingual multilayer perceptron

To train a multilingual multilayer perceptron (ML-MLP) for context-independent phones, we use the knowledge-driven approach to create a universal phone set, i.e., the phone sets of all languages are pooled together and then merged based on their IPA symbols. Afterwards, several training iterations are applied to create the multilingual model and, thereafter, the alignment for the complete data set. Figure 5.1 shows the layout of our MLP architecture which is similar to [MHJ+10]. As the input for the MLP network, 11 adjacent MFCC feature vectors are stacked and the universal phone set is used as the target classes. A 5 layer MLP was trained with a 143-1500-42-1500-X feed-forward architecture, in which X is the number of phones in the universal phone set. In our case, we used the ICSI QuickNet3 software [QN] to train the network. We used a learning rate of 0.008 and a scale factor of successive learning rates of 0.5. The initial values of this network were chosen randomly.

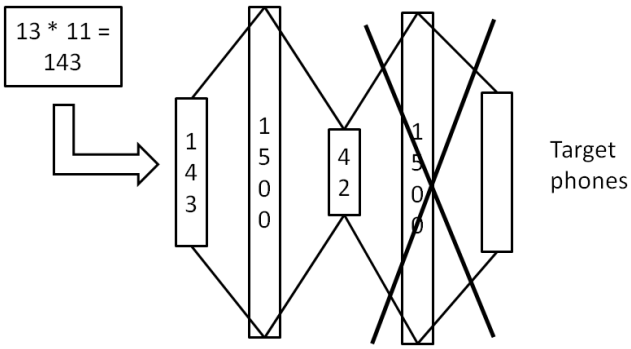


Figure 5.1: Bottle-Neck features

5.3.2 Initialization scheme using multilingual MLP

A multilingual MLP learns to separate the phones in the phonetic space using MFCC features as input. MFCC features are extracted to capture the presentation of the speech signal independent of languages. Moreover, all the languages share a common phonetic space which can be described using IPA (see 2.3). That means, if we have speech data from any new target languages, the multilingual MLP can be used directly without any change to obtain the posterior of each phone in the multilingual phone set. Obviously, if all the phones in the new language are part of the universal phone set, the multilingual MLP can be used directly to classify the phones of the new language. Since the multilingual MLP has never seen the data of the target language, the performance of the multilingual MLP on the data of the target language might not be the best performance which can be achieved. Hence, the idea is to use the multilingual MLP to initialize the MLP training for the new language. By doing so, we obtain a better starting point for optimization compared to randomly generated initial parameters. Figure 5.2 illustrates the initialization scheme. For the new language, we select the output from the ML-MLP based on the IPA table and use it as an initialization of the MLP training. All the weights of the ML-MLP up to the last hidden layer are taken but only the weights and the output biases of the selected targets are used.

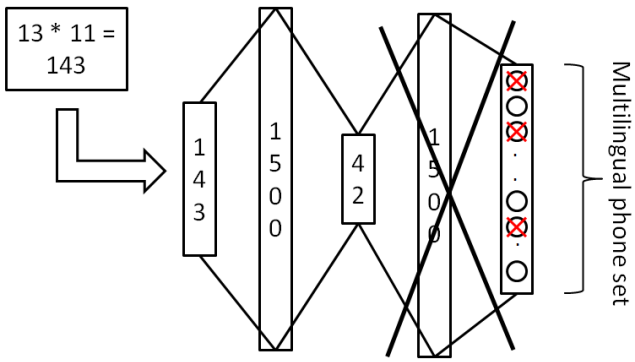


Figure 5.2: Initialization scheme for MLP training or adaptation using a multilingual MLP. Only the phones of the target language are selected.

5.3.3 "Open target language" multilayer perceptron

However, it can be difficult to apply the multilingual MLP to any new language, since even with many languages it is not guaranteed that any new phone in the target language could be covered by the multilingual phone set. In the following example, we use English, French, German and Spanish to train the multilingual acoustic model. The universal phone set has 81 phones which cover only about 30% of all the IPA symbols. This shows that we could encounter difficulties applying this multilingual MLP to a new language, especially, if the amount of training data is limited. So, we propose a new strategy to train an "open target language" MLP network and apply it to language adaptation at feature level. Our idea is to extend the target classes so that we can cover all the phones of the IPA table. Hence first, the training data for the phones which do not appear in the available multilingual training data need to be selected. Since all the phones in IPA are described by their articulatory features, we propose to use the data from several available phones that have the same articulatory features as the uncovered target phone.

For some special phones like aspirated phones or diphthongs, the following steps are applied:

- If the phone is an aspirated phone, use the frames of the begin and middle state of the main phone (e.g. A: A-b, A-m) and the end state of /h/-e.
- If the phone is a diphthong (consisting of two vowels V1 and V2), the frames of V1-b, V1-m and V2-e are used.

To ensure the balance of training data between phones, we randomly choose a subset of the selected data to train the parameters for the new target phones.

After finishing the training data selection for all the new target phones, we first train a usual MLP with a subset of all the training data to save time and learn a rough structure of the phone set which can be covered in our training set. Afterwards, we use this MLP as an initialization and train weights for the new target phones with all the selected data. Due to the fact, that the new target classes are not real, it is possible that the MLP network after this step does not match our real target phones anymore. Hence, we re-train the whole network using all the training data.

5.3.4 Experiments and Results

To evaluate the proposed approach, we conducted the first experiments in which a multilingual MLP with four European languages (English, French, German

and Spanish) was trained. Afterwards, we used it to initialize the MLP training for randomly chosen target languages, in this case Creole and Vietnamese.

Multilingual multilayer perceptron

First, a multilingual MLP was trained with all the English, French, German and Spanish training data using the QuickNet toolkit [QN] which allows neural network training with multi-threading on CPU. The MLP has 5 layers and the topology 143-1500-42-1500-81. For comparison, we also trained different monolingual MLPs with the same topology (only the number of target phones was changed). For all the MLP training, we used a learning rate of 0.008 and a scale factor of successive learning rates of 0.5. Table 5.1 shows the frame-wise classification accuracy for all MLPs using random and multilingual MLP initialization on their cross validation data. The multilingual MLP trained with random values has a frame accuracy rate of 67.61% on its cross validation set which contains English, French, German and Spanish. Using this multilingual MLP, the MLP training for English, French, German and Spanish was initialized and re-trained. We observed overall improvements by using the multilingual MLP as initialization compared to random initialization on the corresponding cross validation set. Moreover, the training was accelerated by up to 40% on average.

Table 5.1: *Frame-wise classification accuracy [%] for all MLPs using random and multilingual MLP initialization on their cross validation data*

Languages	Random Init	Multilingual Init
English (EN)	70.98	73.46
French (FR)	76.73	78.57
German (GE)	63.93	68.87
Spanish (SP)	71.75	74.02

Furthermore, several ASR systems were trained using different BN features for all the languages. The results in Table 5.2 show that BN features improve the baseline system trained with traditional MFCC features for all four languages. Multilingual BN features performed the best in our experiments. In the case of English and German, we observed about 8% relative improvement compared to the BN features.

Table 5.2: *WER [%] on the GlobalPhone development set*

Systems	English	French	German	Spanish
MFCC	11.5	20.4	10.6	11.9
BN	11.1	20.3	10.5	11.6
Multilingual BN	10.2	20.0	9.7	11.2

Language adaptation to Vietnamese

Data selection for MLP training Since not all Vietnamese phones could be covered by the multilingual universal phone set, we had to train several open phones using the multilingual training data. Table 5.3 shows all the uncovered Vietnamese phones and their phonetic features. For uncovered Vietnamese vowels and consonants, we used the training data from the phone with the same articulatory features e.g. Plosive, Palatal for consonant /ch/ or Close, Back for vowel /o3/. For the case of diphthongs such as /ie/, /ua/, and /ua2/, we used the frames of the first two states (-b and -m) of the first vowel and the frames of the last state of the second vowel.

Table 5.3: *Vietnamese phones which are not covered by the universal phone set and their articulatory features*

VN	Articulatory features
/d2/	Plosive, Dental/Alveolar
/tr/	Plosive, Retroflex
/s/	Fricative, Retroflex
/r/	Fricative, Retroflex
/ch/	Plosive, Palatal
/o3/	Close, Back
/ie2/	i-b, i-m, e2-e
/ua/	u-b, u-m, a-e
/ua2/	ir-b, ir-m, a-e

Results For language adaptation experiments, we conducted two different experiments on the Vietnamese GlobalPhone data set. In the first experiment, we used all the training data and trained an ASR system using the BN features. By using random initialization, we achieved 65.13% accuracy on the cross validation set with MLP training and a SyllER of 11.4% on the Vietnamese development set. To obtain a better initialization, we applied the multilingual MLP

from the previous experiment, which led to 67.09% accuracy on the cross validation set and 10% relative improvement in terms of SyllER compared to the MLP system with random initialization.

Table 5.4: *Frame-wise classification accuracy [CVAcc in %] for all MLPs on cross validation data and SyllER [%] from a system trained with 22.5h Vietnamese data*

MLP	CVAcc	SyllER
MFCC	-	12.0
BN	65.13	11.4
Multilingual BN	67.09	10.1

In the second experiment, we assumed to have only a small amount of training data (about 2 hours) for Vietnamese. We trained the baseline system using MFCC features and observed a SyllER of 26% on the Vietnamese development set. Since two hours are not enough for MLP training, we directly used the multilingual MLP which was trained in the previous experiment to extract the BN features without any re-training. The SyllER was improved by 0.7% absolute which indicates that useful, language independent information has been learned during the MLP training. To perform a comparison with our new approach, we adapted the MLP with 2h of Vietnamese data using the approach in [TFGK08] when the hidden-to-output weights and output biases were initialized randomly. The advantage of this approach is that no manual phone mapping needs to be provided. The results were improved significantly (by about 20% in terms of cross validation accuracy and 2.5% absolute in terms of SyllER). After that, we applied the proposed multilingual Bottle-Neck features, in which we used all the weights and output biases of the multilingual MLP. We observed 0.8% absolute improvement after adaptation in MLP training and 1.2% absolute improvement in terms of SyllER. It indicates that the last softmax layer also contains some language independent information which can be transferred between languages.

MLP initialization using monolingual MLP vs. multilingual MLP

The success of our experiments described in the last sections raises an important question: Do we need a multilingual MLP to initialize the MLP training for a new language? Or is it enough to use a monolingual MLP? Therefore, we conducted experiments on Haitian Creole in which we compared different initialization schemes for MLP training: random initialization, using monolingual, and multilingual MLP, and their impact on the ASR system. We chose

Table 5.5: *Frame-wise classification accuracy [CVAcc in %] for all MLPs on cross validation set and SyllER [%] from a system trained with 2h Vietnamese data*

Systems	CVAcc	SyllER
MFCC	-	26.0
ML-MLP	37.23	25.3
Adapted ML-MLP	57.54	22.8
Multilingual BN	58.32	21.6

French (FR) for the monolingual MLP since Haiti Creole is related to French. We applied our approach to train the "open target language" MLP with only 80 hours French data from the BREF database [LGE+91] (*Monolingual-BN*) and used it for the MLP training for Haiti Creole. Furthermore, we also applied the ML-MLP trained in 5.3.4 to initialize the MLP training. Table 5.6 shows the frame-wise classification accuracy for all the MLPs trained with different initializations on cross validation data and their WER on the Creole data set. Using the MLP trained with French data for initialization, we observed a small

Table 5.6: *Frame-wise classification accuracy [CVAcc in %] for all MLPs on cross validation data and WER [%] on Creole database*

Systems	CVAcc	WER
Baseline (MFCC)	-	12.3
BN (random init)	73.36	11.6
Monolingual-BN	75.15	11.4
Multilingual-BN	75.38	10.4

improvement in terms of WER (0.2% absolute), but the final performance is still worse than the system trained with multilingual MLP initialization which gave 1.9% absolute improvement.

Robustness against transcriptions errors

In this paragraph, the robustness of our proposed approach is verified. We applied the multilingual MLP to initialize the MLP training for Vietnamese in which the audio data contain transcription errors. Using our multilingual unsupervised training framework - MUT - (as proposed in Chapter 4), we built a Vietnamese ASR with 4 different source languages (English, French, German and Spanish). In total, 10 hours of training data with automatic transcriptions

which have 16% SyllER could be selected. Based on these transcriptions, the baseline system using MFCC features has 18.6% SyllER on the evaluation set. Afterwards, we trained two different ASR systems using Bottle-Neck features to improve accuracy: one using random initialization and another one using the multilingual MLP trained in 5.3.4. Table 5.7 shows the frame-wise classification accuracy for all the MLPs on cross validation data and the SyllER from all the systems trained with MUT. The results indicate that initializing an MLP training with random values can be problematic for the case of automatically transcribed data (SyllER increases 0.4% absolute) while using the multilingual MLP as initialization is much more robust (2.0% absolute improvement).

Table 5.7: *Frame-wise classification accuracy [CVAcc in %] for all the MLPs on cross validation data and SyllER [%] from all the systems trained with our Multilingual Unsupervised Training Framework*

Systems	CVAcc	SyllER
MFCC	-	18.6
BN (random init)	61.5	19.0
Multilingual-MLP	65.0	16.6

5.4 Multilingual multilayer perceptron for rapid language adaptation between and across language families

In this section, we present our investigations of multilingual multilayer perceptrons (MLPs) for rapid language adaptation between and across language families. We explore the impact of the amount of languages and data used for the multilingual MLP training process on the final ASR performance. Furthermore, we aim at finding the effect of the similarity between source and target languages on the MLP performance and the corresponding ASR performance. In total, two different experiments were conducted: using all the training data and using only a small amount of training data of the Czech, Hausa, and Vietnamese GlobalPhone data set. In both cases, we applied different multilingual MLPs for the MLP training initialization and also experimented with and without re-training.

5.4.1 Experimental setup

For this research, we selected French, German, Spanish, Bulgarian, Polish, Croatian, Russian, Czech, Portuguese, Mandarin, Korean, Thai, Japanese, Hausa and Vietnamese from the GlobalPhone corpus. In addition, we used English speech data from WSJ0. We used Czech, Hausa, and Vietnamese as target languages and the remaining ones as source languages. We split the source languages into three different categories in order to perform the experiments: The first one called *Big4* contains European, resource-rich languages like English, French, German, and Spanish. The second one consists of four different *Slavic* languages, namely Bulgarian, Croatian, Polish and Russian. The last one is composed of the four *Asian* languages Chinese, Japanese, Korean and Thai, in which Chinese and Thai belong to the *Sino-Tibetan* language family and Korean and Japanese are from the *Altaic* language family. However according to different linguistic studies, Korean can also be classified as language isolate, i.e. having no relationship to any other languages.

5.4.2 Rapid language adaptation for new languages

In the first experiment, we applied different multilingual MLPs to initialize the MLP training and used all the training data to train the monolingual MLP for each target language. Table 5.8 shows the frame-wise classification accuracy for all the MLPs trained with different initializations on cross validation data. Note that the number of the MLP outputs is the number of the phones from the target language. All the MLPs for Czech, Hausa and Vietnamese have 42, 34, and 39, respectively. We observed a significant improvement over the MLP trained with random initialization. As we increased the number of source languages and the amount of data to train the multilingual MLP, the final performance of the target language MLP on the cross validation set increases slightly. However, we did not observe any impact of using related source languages on the MLP performance. For all three target languages, the best MLP performance was obtained by using *Big4* MLP to initialize the MLP training.

After finishing the MLP training, all the MLPs were used to extract the BN features for the ASR experiments. Table 5.9 shows the ASR performance for Czech, Hausa, and Vietnamese with MFCC features and BN features which were initialized with different multilingual MLPs. Note that the multilingual MLPs were trained on speech data from the same and from different language families compared to the target language. The results show overall improvements of ASR performance compared to the MFCC and the MLP with random initialization even if the source languages and the target language are not in

Table 5.8: *Frame-wise classification accuracy [%] of the target language MLPs with different initializations on cross validation data*

Initialization	Czech	Hausa	Vietnamese
Random	72.34	73.47	65.13
Big4 (4 languages)	76.62	76.49	67.09
Slavic (4 languages)	76.28	76.38	66.94
Asian (4 languages)	76.05	76.61	67.05
Big4 + Slavic (8 languages)	77.13	76.70	67.56
Big4 + Slavic + Asian (12 languages)	77.62	76.92	68.08

the same language family. However, for the case of Czech, we obtained significantly better results by using the *Slavic* source languages which are from the same language family as the target language. Vietnamese ASR obtained the best results by using *Asian* MLP, however, the difference in terms of SyllER between using *Big4* and *Asian* is very small. Furthermore, it is difficult to draw a conclusion for Vietnamese, since only two source languages namely Chinese and Thai are from the same language family - *Sino-Tibetan* - as Vietnamese. In the case of Hausa, the word error rate is almost independent of the multilingual MLP which was used for the initialization process. The results in table 5.8 and 5.9 indicate that there is no correlation between the MLP performance on the cross validation set and the final ASR performance.

Table 5.9: *ER [%] for Czech, Hausa, and Vietnamese ASR using MFCC features and BN features with different multilingual MLPs between and across language families for initialization*

Systems	Czech	Hausa	Vietnamese
MFCC	19.5	14.6	12.1
BN	19.2	15.4	11.4
Big4 (4 languages)	16.8	14.2	10.1
Slavic (4 languages)	**16.3**	14.2	10.7
Asian (4 languages)	17.1	**14.1**	**10.0**

In the next experiments, we successively increased the number of languages and, therefore, obviously the amount of data to train different multilingual MLPs which we used to initialize the MLP for our target languages. Figure 5.3 illustrates the ASR performance on Czech, Hausa, and Vietnamese test data using those different BN features. The results show that the more languages and the more data we used to train the multilingual MLP, the better was the final ASR performance. The improvements tended to be larger, especially if

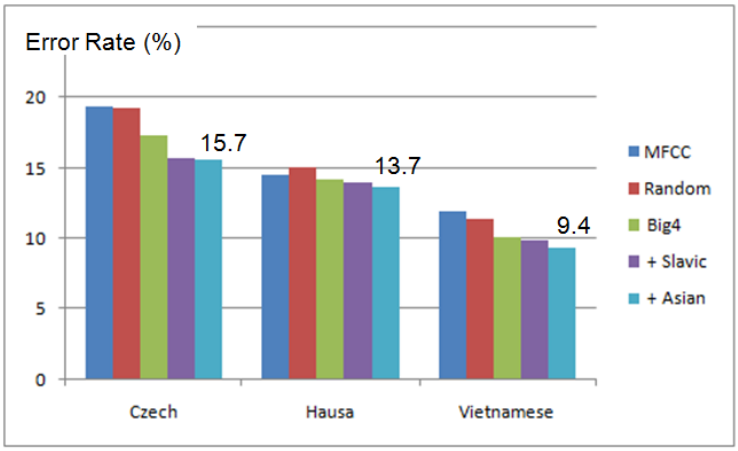


Figure 5.3: ER for Czech, Hausa, and Vietnamese ASR trained on all the training data using MFCC features, and BN features with different initializations

the source languages and the target language were in the same language family. In the case of Czech, the WER dropped from 16.8% to 15.8% when we added all four *Slavic* languages in addition to the *Big4* source languages. Afterwards, although the four *Asian* source languages were added, i.e. we increased the amount of languages and also the data, the WER was improved only very slightly. In contrast, in the case of Vietnamese, when we added the four *Asian* languages, the WER was improved more than by adding *Slavic* languages, since the *Asian* group contains two *Sino-Tibetan* languages as Vietnamese. The results indicate that adding related languages into the set of the source languages to train the multilingual MLP has a strong effect on the ASR performance of the target language. For the case of Hausa, we also observed improvement even if all the source languages are very different from the Hausa language based on the language families.

5.4.3 Rapid language adaptation for low-resource languages

In the second experiment, we assumed to have very little training data (about 10% of the full training data) for Czech, Hausa, and Vietnamese. We trained the baseline system using MFCC features and obtained an ER of 27.5%, 24.9% and 26% on the Czech, Hausa, and Vietnamese test set respectively. Since two hours training data are not enough for an MLP training, we directly used the

multilingual MLPs which were trained in the previous experiment to extract the Bottle-Neck features. We also trained an Oracle system for each target language by using the best MLP which was trained with the full training data from the previous experiments. Figure 5.4 illustrates the ASR performance for Czech, Hausa, and Vietnamese using different multilingual MLPs. Again, the more languages and the more data we used to train the multilingual MLP, the better was the final ASR performance. In contrast to our experiments with the full amount of data, we observed substantial improvements every time we added more data of other languages to train the multilingual MLP. The results indicate that if only a very small amount of training data of the target language is available, the impact of adding more languages and more data is stronger on the ASR performance than the relativeness between source and target languages. However, the best performance in the case of Hausa and Vietnamese is rather far away from the Oracle result, but not for Czech. Since the ASR performance increases almost proportionally with the number of languages used to train the multilingual MLP, it seems to be very promising to achieve similar results to the oracle experiments with more languages.

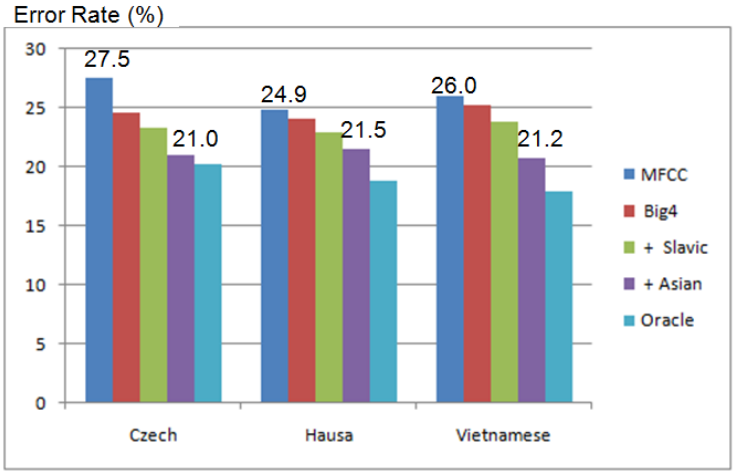


Figure 5.4: ER for Czech, Hausa, and Vietnamese ASR trained on a very small amount of training data using MFCC features, and BN features with different initializations without re-training

Furthermore, we also re-trained the multilingual MLP using the available data of the target language to improve the MLP accuracy. Table 5.10 presents the frame-wise classification accuracy of the target language MLPs with different initializations on cross validation data after re-training. We observed a con-

sistent improvement on the MLP performance by adding more training data from other languages to train the multilingual MLP. It is notable to observe that even if the source languages and the target language are not related, we still obtained additional gain on the MLP performance. Using the BN features

Table 5.10: *Frame-wise classification accuracy [%] of the target language MLPs with different initializations on cross validation data*

Initialization	Czech	Hausa	Vietnamese
Big4 (4 languages)	70.58	71.12	58.32
Big4 + Slavic (8 languages)	72.18	72.56	60.12
Big4 + Slavic + Asian (12 languages)	72.38	73.42	62.38

extracted from the re-trained MLP, we re-trained the AM and observed an overall improvement compared to the system without MLP re-training. In average, an improvement of around 4% relative was obtained. Table 5.11 summarizes the ER for Czech, Hausa, and Vietnamese ASR using MFCC and BN features with different multilingual MLPs for initialization after re-training.

Table 5.11: *ER [%] for Czech, Hausa, and Vietnamese ASR using MFCC features, and BN features with different initializations after re-training*

Systems	Czech	Hausa	Vietnamese
MFCC	27.5	24.9	26.0
Big4	23.8	23.7	22.8
+ Slavic	22.0	22.4	21.7
+ Asian	**20.9**	**21.3**	**20.3**
Oracle	20.2	18.8	18.0

5.5 Visualization of Bottle-Neck features

For a better understanding of the multilingual Bottle-Neck features, we visualized them in a two-dimensional space. To reduce the data dimension of the multilingual BN features to 2D, we applied t-Distributed Stochastic Neighbor Embedding (t-SNE) [VdMH08].

5.5.1 t-Distributed Stochastic Neighbor Embedding

Visualization of high-dimensional data is an important task in many different domains, and has to deal with data of widely varying dimensionalities. Over the last few decades, a variety of techniques for the visualization of such high-dimensional data have been proposed. One of the latest techniques which works quite well in many applications is t-Distributed Stochastic Neighbor Embedding (t-SNE) [VdMH08] - an extension of Stochastic Neighbor Embedding [HR02]. It is a technique which allows visualizing high-dimensional data by assigning each data point a location in a two or three-dimensional space.

Stochastic Neighbor Embedding (SNE) starts by converting high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities. The similarity of data point x_j to data point x_i is the conditional probability $p_{j|i}$ that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i. For the low-dimensional counterparts y_i and y_j of the high-dimensional data points x_i and x_j, a similar conditional probability $q_{j|i}$ is computed. If the mapped points y_i and y_j correctly model the similarity between the high-dimensional data points x_i and x_j, the conditional probabilities $p_{j|i}$ and $q_{j|i}$ will be equal. Based on this observation, SNE aims at finding a low-dimensional data representation that minimizes the mismatch between $p_{j|i}$ and $q_{j|i}$. A natural measure for that is the Kullback-Leibler divergence. SNE minimizes the sum of Kullback-Leibler divergences over all the data points using a gradient descent method. The cost function C is given by

$$C = \sum_i KL(P_i|Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}} \tag{5.1}$$

in which P_i represents the conditional probability distribution over all other data points given data point x_i, and Q_i represents the conditional probability distribution over all other map points given map point y_i. Although SNE constructs reasonably good visualizations, the cost function is difficult to optimize. Also, the authors in [VdMH08] refer to the "crowding problem", which t-SNE tries to alleviate. The cost function used by t-SNE differs from the one used by SNE in two ways: (1) it uses a symmetrized version of the SNE cost function with simpler gradients and (2) it uses a Student-t distribution rather than a Gaussian to compute the similarity between two points in the low-dimensional space. The t-SNE software is provided in [t-S] and used in our further experiments.

5.5.2 Visualization

In this section, we applied t-SNE to visualize the multilingual BN features. We hope to find answers to the following questions:

- What does the multilingual MLP learn?
- Does the BN representation transfer to new languages?

The following paragraphs discuss the visualization of the multilingual BN features and possible implications.

What does the multilingual MLP learn? To extract the BN features, we used the multilingual MLP which has been trained on 12 different languages (see Section 5.4). Since the number of phones of a language is too large for the visualization, a subset of phones is selected. In this thesis, we only focus on visualizing vowels. We chose five different vowels /a/, /i/, /e/, /o/, and /u/ which are covered in many languages. We plotted the multilingual BN features of these five vowels /a/ (black), /i/ (blue), /e/ (green), /o/ (red), and /u/ (yellow) (on the right in Figure 5.5). The data points were collected by using French (+), German (□) and Spanish (▽) speech data. On the left of Figure 5.5, we show the IPA vowel chart and the vowel-triangle with the five vowels annotated with corresponding colors. Note that the vowel-triangle expresses which vowels have which formants on average. Interestingly, an analogy of the visualization with the other two pictures can be observed. The data points of the five vowels from the four different languages resemble the relations of the vowels in the vowel chart and the vowel-triangle. This observation suggests the following implications:

- An MLP captures important information about the vowel realizations. It has learned spectral characteristics of different vowels, namely the first two formants *F1* and *F2*. According to our results, t-SNE allows to visualize that the MLP learned to discriminate different vowels and abstracts from languages.
- An MLP seems to normalize the language dependent variations of these vowels. Although the data points are from different languages, they clearly resemble the pattern of the IPA vowel chart and the vowel-triangle.

Does the BN representation transfer to new languages? As described in Section 5.4.3, we obtained significant improvements in terms of SyllER by using the multilingual MLP directly without re-training to extract the BN features

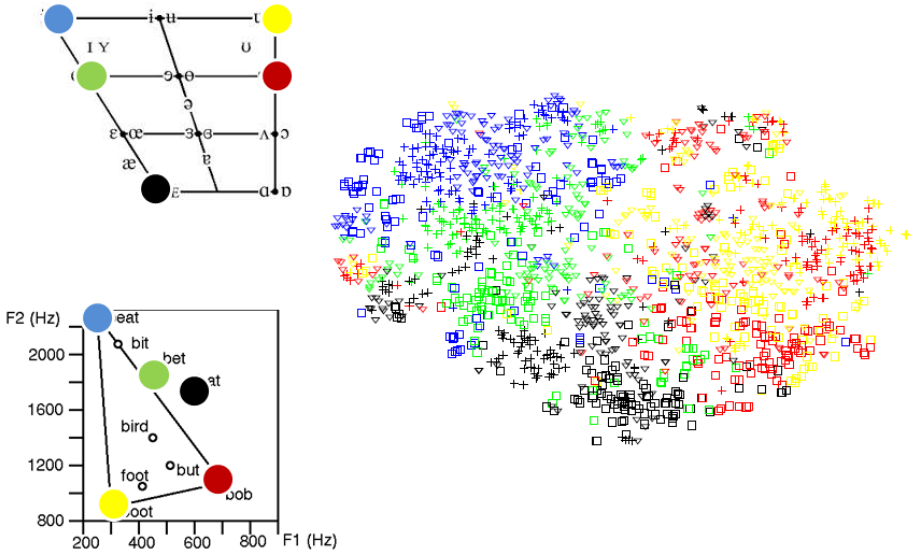


Figure 5.5: Multilingual BN features of five vowels /a/ (black), /i/ (blue), /e/ (green), /o/ (red), and /u/ (yellow) from French (+), German (□) and Spanish (▽)

for Vietnamese ASR. This indicates that some language independent information has been learned by training the multilingual MLP. However, it was not clear how exactly the language independent information is represented in this context. In the previous paragraph, we observed that the multilingual MLP captures the most important information of the vowels, namely *F1* and *F2* and normalizes language variations. This can be the explanation for the ASR performance improvement reported in Section 5.4.3.

In this section, we visualize the BN features of Vietnamese data using this multilingual MLP to obtain a better understanding of the crosslingual transfer effect. Moreover, we look at two further effects: The language independence of the BN features and the discriminability of the multilingual BN features for unseen languages. The intra-class variance of vectors from different languages for the same IPA symbol is observed. In particular, we plotted the five vowels which appear in German, French, Spanish and Vietnamese. Note that German, French and Spanish data was used to train the multilingual MLP while Vietnamese is the unseen language in our example. Figure 5.6 shows the multilingual BN features of /a/, /e/, /u/, /i/ and /o/, respectively. In this figure, data points are color coded corresponding to German (red), Spanish (black), French (purple) and Vietnamese (yellow) phones.

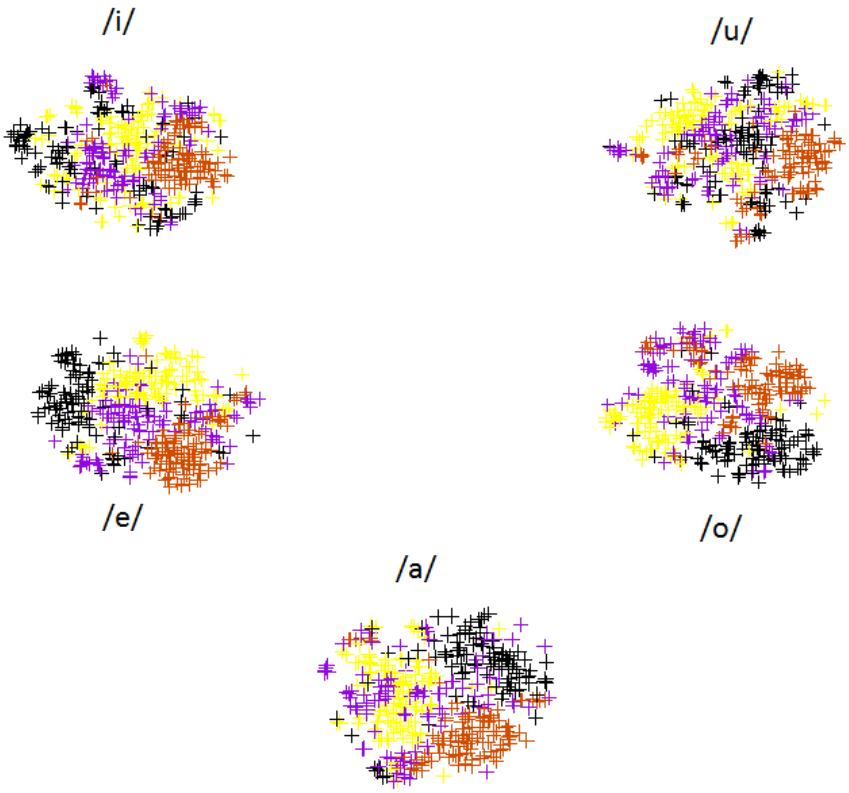


Figure 5.6: BN features of five vowels /a/, /i/, /e/, /o/, and /u/ from German (red), Spanish (black), French (purple) and Vietnamese (yellow)

We observed two characteristics:

- The data points form a compact class even if they are from different languages.
- There exists an overlap of data points from different languages. This indicates that the intra-class variance of each class is small.

These two observations indicate that multilingual BN features may be language independent. However based on these figures, it is not possible to conclude whether multilingual BN features are also suitable to the classification task since we only plotted the data points of one class. Therefore, multilingual BN features of the five Vietnamese vowels are extracted and plotted on the right

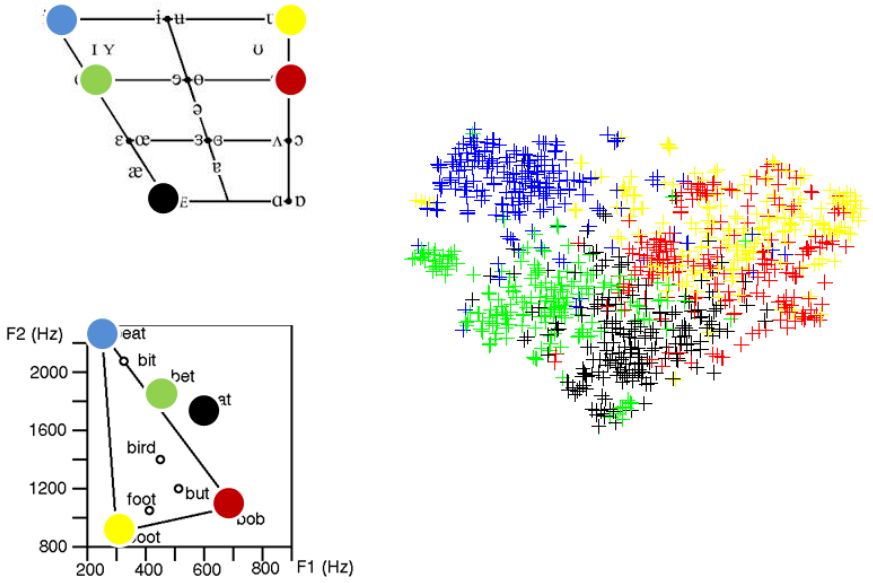


Figure 5.7: BN features of the five Vietnamese vowels /a/ (black), /i/ (blue), /e/ (green), /o/ (red), and /u/ (yellow) using multilingual MLP trained with 12 different languages 5.4

hand side of Figure 5.7. Black, blue, green, red and yellow data points correspond to the vowels /a/, /i/, /e/, /o/, and /u/. On the left hand side of Figure 5.7, we show the vowel chart and the vowel-triangle again. Interestingly, we observed the same effect as by visualizing the multilingual phones in Figure 5.5. The data points of the five Vietnamese vowels again represent the relations in the vowel chart and the vowel-triangle. This indicates that the learned information, in this case the *F1* and *F2* information, can be transferred to the new language. This means, the multilingual BN features are language independent and can be used for feature extraction for an unseen language.

The next question is: How important is the use of multilingual MLP or is it enough to use a monolingual MLP? To answer this question, we again plotted the BN features of the same Vietnamese vowels as in Figure 5.7. However, in this case only a monolingual MLP was used to extract the features: A French MLP trained on French GlobalPhone data with random initialization. Figure 5.8 illustrates the IPA vowel chart and the vowel-triangle on the left and on the right the Vietnamese data points. Note that, Vietnamese data was not used for the MLP training. Again, the same effect as in Figure 5.5 and 5.7 is observed. The data points of the five Vietnamese vowels illustrate the relations

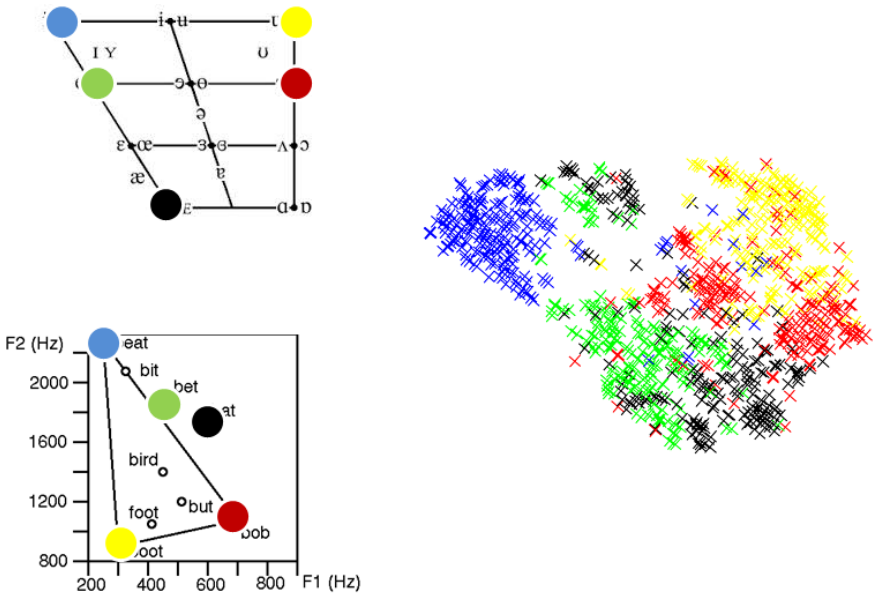


Figure 5.8: BN features of five Vietnamese vowels /a/ (black), /i/ (blue), /e/ (green), /o/ (red), and /u/ (yellow) using MLP trained with French data

in the vowel chart and the vowel-triangle. It indicates that the MLP learned the spectral characteristics, namely *F1* and *F2* of different vowels. It can be transferred to an unseen language independent of whether monolingual or multilingual data are used to train the MLP. However, the analogy between the pattern in the plotted data points and the vowel charts in Figure 5.7 is more clear than in Figure 5.8. It can be observed in Figure 5.8 that some data points of phone /a/ and /e/ are spread and form a pattern close to phone /i/. One possible explanation for this effect is that the more languages and more data are used to train the MLP, the stronger is the normalization process between languages at the phone level. It also explains the ASR performance which we obtained in Section 5.3.4: Using the French MLP for initialization, the ASR performance was improved, but the final performance was substantially worse than the system trained with multilingual BN features.

Furthermore, we plotted Vietnamese BN features in Figure 5.9 which have been extracted using the Vietnamese MLP trained on Vietnamese data with random

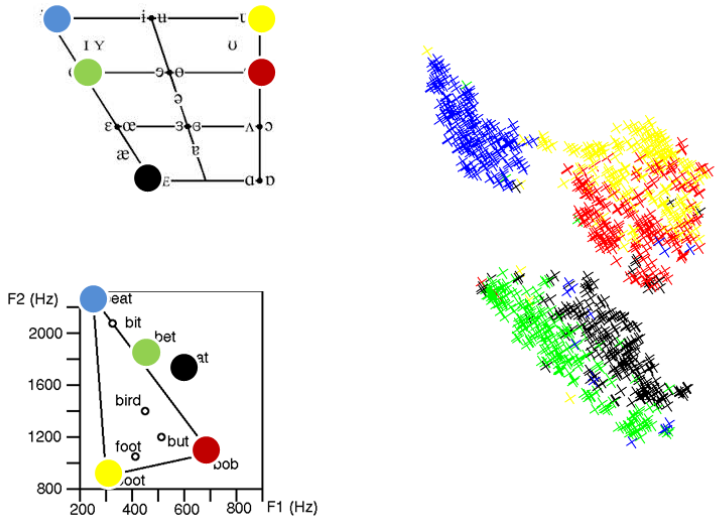


Figure 5.9: BN features of five Vietnamese vowels /a/ (black), /i/ (blue), /e/ (green), /o/ (red), and /u/ (yellow) using MLP trained with Vietnamese data

initialization. We also observed an analogy between the vowel chart and the data points. Furthermore, in comparison to the visualization in Figure 5.7 and 5.8, the data points of different vowels are clearly separable. It proves the discriminative characteristics of the MLP training process. In this case, it was optimized to separate between Vietnamese phones. It also indicates that MLP training is more effective when trained on the target language. However in scenarios with limited training data, using our multilingual MLP to initialize the MLP training is a good way to train the MLP for a new language.

5.6 Summary

This chapter presented our investigations on multilingual Bottle-Neck features and their application to rapid language adaptation to a new language at feature level. Our results revealed that using the multilingual MLP to initialize the MLP training for new languages improved the MLP performance and, therefore, the ASR performance. Figure 5.10 summarizes the ASR performance on 15 languages of the GlobalPhone test set using the proposed multilingual Bottle-Neck features. The ASR performance was improved in all the cases in compar-

ison to the results with MFCC features which are presented with blue bars in Figure 5.10 and which were also shown previously in Figure 3.1.

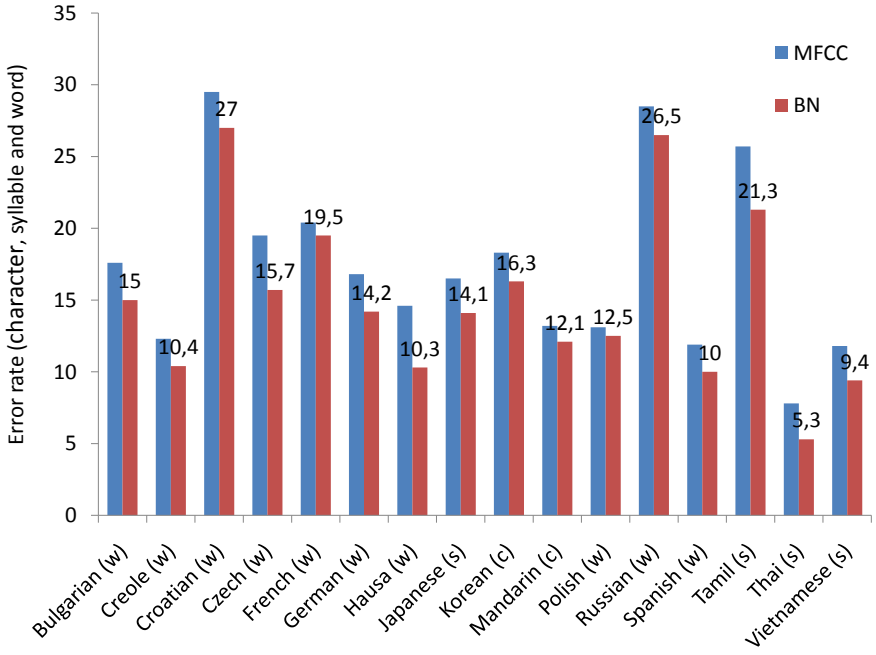


Figure 5.10: ASR performance on the GlobalPhone test set using multilingual Bottle-Neck features (c: character, s: syllable, w: word)

Moreover, we investigated the impact of the source languages on the MLP training and the ASR performance of the target languages. The experimental results showed that the number of languages and the amount of data used to train the multilingual MLP has a strong effect on the MLP training and the ASR performance. More source languages improve the MLP performance for a new language and the ASR performance. Moreover, depending on the amount of training data of the target languages, language relation between source languages and target languages becomes important. If many training data are available, it helps to use related languages. In contrast, if only a small amount of training data is available, language similarity does not help but the number of source languages and the amount of data matters. Multilingual Bottle-Neck features are language independent and can be used for rapid language adaptation without re-training to improve the ASR performance. However, even with

a very small amount of training data (one hour of data in our experiment), MLP re-training helps to improve the ASR performance. Finally, the visualization of the output of the hidden layer of the MLP using t-SNE provides useful information to better understand the multilingual BN features. Our results revealed that multilingual BN features seemed to learn the *F1* and *F2* formants which characterize different vowels and normalized their language dependent variations. Furthermore, the BN features representation transferred to unseen languages which further indicates their language independence.

CHAPTER 6

A Study on Using Multilingual and Crosslingual Information To Improve Non-Native ASR

Non-native speech is still a challenging task for state-of-the-art ASR systems. The word error rate increases significantly on testing data with foreign accents. This chapter presents the exploration of the effect of using multilingual and crosslingual information to improve an ASR system for non-native speech.

6.1 Introduction

Another advantage of multilingual systems compared to monolingual systems is their application to non-native speech recognition. For state-of-the-art ASR systems, non-native speech is a challenging task. There are many reasons why an automatic speech recognition (ASR) system which performs well on native speakers has problems with non-native speech. Two of them are the characteristics of accented speech itself and the lack of speech databases. In [Liv99,

TW03], some of the speaker-related factors that have negative impact on speech recognition performance for non-native speech are presented, such as:

- High intra- and inter-speaker inconsistencies of the phonetic realizations
- Different second language acquisition methods and backgrounds, thus different acoustic or grammatical realizations and proficiency levels,
- The speakers' perception of the non-native phones
- Reading errors in read speech
- Slower reading with more pauses in read speech.

Due to the high variations among speakers, a large amount of training data is required to build a robust acoustic model for non-native speech. However, obtaining those training data is very difficult, especially for speakers with strong accent. Hence, the use of multilingual acoustic models is investigated in this thesis to increase the robustness of the model against accent variations, compensate data spareness and, therefore, improve the ASR performance on non-native speech.

In this chapter, we explore the use of multilingual and crosslingual information in different ways. We will use the terms L1 to refer to the native languages of the speakers, and L2 to refer to the language that the ASR system is trained to recognize. We investigate the effect of using a bilingual acoustic model which was trained with L1 and L2 data on non-native speech. For the case that L1 is unknown or the data of L1 is not available, a multilingual acoustic model trained without L1 training data is examined. Furthermore, for scenarios, where no adaptation data is available, we propose a new method called crosslingual accent adaptation which allows, for example, using English with Chinese accent to improve the German ASR on German with Chinese accent.

6.2 Related work

There are many previous research works on handling non-native speech in speech recognition. The investigations vary from simply collecting data in the target accent and training new acoustic models, to various ways of adapting pronunciation dictionary, acoustic model, and language model to the new accent.

In [WSW03], different techniques that improve the recognition performance for non-native speech are compared. The study uses spontaneous German-accented English and investigates different approaches, such as using a bilingual acoustic model, a model built from mixed (native and non-native) speech,

maximum a posteriori (MAP) speaker adaptation, acoustic model interpolation, and polyphone decision tree specialization. The authors obtained a great improvement on German-accented speech but did not achieve any substantial improvements using bilingual acoustic models. Tomokiyo and Waibel [TW03] examined Japanese-accented English speech and showed that training on non-native speech data achieves the biggest gains in performance on accented data. In both cases, the adaptation was based on the direct use of MAP or maximum likelihood linear regression (MLLR) to adapt to each test speaker individually or to a class of accented speakers. In [RGN08, TB07], the authors applied multilingual weighted acoustic models to improve recognition accuracy for non-native speech recognition. Bouselmi et. al [BFI$^+$06] showed a great improvement by modifying the acoustic model using phonetic confusion rules which have been extracted from a non-native speech database for a given L1 and L2 using both the ASR systems of L1 and L2. The results in [RGN08, TB07, BFI$^+$06] indicate that there is some multilingual information which might be useful to improve ASR performance on non-native speech.

Beside acoustic model adaptation, there are also many works on modifying the decoding dictionary so that it reflects the pronunciation differences between native and non-native speech, such as in [Liv99, Tom00b, GE03, HWP96]. Moreover, the language model can be adapted to non-native speech [TW03]. However, adapting the pronunciation dictionary or the language model do not form the focus of the research in this thesis.

6.3 Baseline System

This section describes the English and the German baseline recognizers. The English system serves as baseline in the experiments in Section 6.4, while the German system is used as baseline in Section 6.5. They can be described as follows: Each system uses Bottle-Neck front-end features with a multilingual initialization scheme as proposed in Chapter 5. In this approach, a multilingual multilayer perceptron (ML-MLP) was trained using training data from 12 languages (Bulgarian, Chinese Mandarin, English, French, German, Croatian, Japanese, Korean, Polish, Russian, Spanish, and Thai). To initialize the MLP training for the English and German system, we selected the output from the ML-MLP based on the IPA phone set and used it as starting point for the MLP training. All the weights from the ML-MLP were taken but only the output biases from the selected targets were used. To rapidly bootstrap the system, the phone models were seeded by their closest matches of the multilingual phone inventory MM7 [SW01b] derived from an IPA-based phone mapping. The acoustic model used a fully-continuous 3-state left-to-right Hidden-

Markov-Model. The emission probabilities were modeled by Gaussian Mixtures with diagonal covariances. For context-dependent acoustic models, we trained a quintphone system and stopped the decision tree splitting process at 2,500 leaves. After context clustering, a merge&split training was applied, which selects the number of Gaussians according to the amount of data. For all the models, we used one global semi-tied covariance (STC) matrix after a Linear Discriminant Analysis (LDA). The language model was built with a large amount of text data crawled with the Rapid Language Adaptation Toolkit [RLA12]. The vocabulary size of the English language model is 60k. Table 6.1 summarizes the perplexity, and out-of-vocabulary rate (OOV) of the English and German language model on the native and non-native test set of English and German respectively. We only report one PPL and OOV rate for the non-native English test sets since the read text is the same for all accents.

Table 6.1: *PPL and OOV of the language model*

Set	3-gram PPL	OOV
Native EN test set	274	0.3
EN with non-native accent test set	121	0.05
Native GE test set	552	0.3
GE with non-native accent test set	433	0.06

The vocabulary size of the German language model is 37k. On the native German and German with Chinese accent test set, the perplexity is 552 and 433, and the OOV is 0.3% and 0.06%, respectively.

The English and German ASR obtained a word error rate (WER) of 9.4% and 14.3% on the native data set, respectively. On the non-native speech data set, our baseline ASR performance varies among 60.0% WER on English data with Bulgarian accent, 57.6% with Chinese accent, 62.2% with German accent, 67.5 % with Indian accent and 59.6% on German data with Chinese accent. Since the acoustic conditions of the native and non-native corpus are quite similar, we assume that the highly drop of WER from the native to non-native speech test set is due to a phonetic mismatch between non-native and native speech.

We applied MAP and MLLR to our baseline system for each accent to improve the ASR accuracy. Table 6.2 provides an overview of our baseline system on English with different non-native accents with and without adaptation. The results show that, using MAP adaptation we gained a lot of improvements over the baseline system and much more than using MLLR. The combination of MLLR and MAP gives the best performance on English with Bulgarian and Indian accent. Furthermore, the best WER after adaptation on German data with Chinese accent is 43.2%.

Table 6.2: *Word error rates (WER) on English with non-native accents using a monolingual acoustic model*

Accents	BG	CH	GE	IN
English ASR (1)	60.0	57.6	62.2	67.5
(1) + MAP	43.1	**38.4**	**43.1**	36.1
(1) + MLLR	49.6	46.2	51.7	48.7
(1) + MAP + MLLR	**43.0**	41.4	43.6	**33.1**

6.4 Improving ASR performance on non-native speech using multilingual information

6.4.1 Bilingual L1-L2 acoustic model

Many previous studies [BFI+06, Fle80, Fle87, DC97, FFN97] showed that the native language L1 has an impact on the pronunciation of L2. Therefore, it is reasonable to use not only L2 but also L1 audio data to train the acoustic model which covers the L1 and L2 phonetic space and, therefore, improves the ASR performance. Hence, we train a bilingual acoustic model for each accent using English data of WSJ0 and data from the native language in the GlobalPhone database. We merge all the phones which share the same symbol in the IPA table and apply the same training procedure as for the training of the baseline system. To model more contexts, we increase the number of leaves of the decision tree to 3,000 quintphones. Table 6.3 shows the WER of the bilingual models on non-native test data. The results show improvements up to 27% for all accents. On top of the bilingual acoustic models, we applied MAP, MLLR and their combination for adaptation. Similar to the experiments of the baseline system, using MAP gained much more improvement than MLLR. However, in contrast to the baseline system, the combination of MLLR and MAP consistently gives some improvements in terms of word error rate for all the accents. The reason can lie within the fact that our bilingual L1-L2 acoustic model was trained with more training data and, therefore has more Gaussians than the monolingual baseline system. Hence, many Gaussians might not be adapted using MAP adaptation but might be transformed by MLLR adaptation.

6.4.2 Multilingual acoustic model

In many cases, information about L1 or L1 data is not available. The question here is whether multilingual information still helps. Hence, we train four different multilingual AMs for each accent in which we omit the L1 speech data.

Table 6.3: *Word error rates (WER) on English with non-native accent using bilingual acoustic models*

Accents	BG	CH	GE	IN
English ASR	60.0	57.6	62.2	67.5
Bilingual L1-L2 ASR (2)	53.2	52.2	45.3	60.2
(2) + MAP	38.4	34.3	36.8	34.0
(2) + MLLR	43.3	41.1	41.7	45.3
(2) + MAP + MLLR	**37.6**	**34.1**	**36.5**	**31.8**

For English with German accent, for example, a multilingual AM is trained on English, Mandarin, Bulgarian, and Indian speech data. Table 6.4 summarizes the WER on the test sets of our four different accents. Compared to the monolingual system, we observe improvements in all cases. Except for the case of Indian accent, the WER is worse than using the bilingual L1-L2 acoustic model even if the number of parameters of the multilingual acoustic model is higher than the corresponding bilingual L1-L2 acoustic model. It indicates that L1 has a strong effect on L2 and, therefore, we can improve the ASR performance by using L1 speech data. However, we achieved the best WER on English with Indian accent with 29.6% by using a multilingual acoustic model trained with Bulgarian, Chinese, German and English data. It corresponds to about 7% relative improvement over the bilingual L1-L2 AM. The reason could lie within the fact that the multilingual acoustic model trained with four different languages might cover more variations in the phonetic space than the monolingual and also the bilingual English-Tamil acoustic model. Since English with Indian accent has a lot of variations, it might benefit more than other accents from using this multilingual model. Although it is not clear whether the improvement is due to the amount of training data or the multilingual effect, the results show that non-native data has a lot of phonetic variations. They cannot be covered by using only monolingual AM trained with L2 speech data. Hence, the results demonstrated the advantages of the multilingual acoustic model over the monolingual one.

6.5 Crosslingual accent adaptation

The approaches described in the previous sections rely on the availability of L2 speech data to adapt the background model. In this section, we describe a method called *crosslingual accent adaption* which can be applied when no such data is available.

Table 6.4: *Word error rates (WER) for English with non-native accents using multilingual acoustic models*

Accents	BG	CH	GE	IN
English ASR	60.0	57.6	62.2	67.5
Bilingual L1-L2 ASR	53.2	52.2	45.3	60.2
Multilingual ASR (3)	54.0	49.4	51.1	50.8
(3) + MAP	42.0	37.4	39.7	32.3
(3) + MAP + MLLR	**41.6**	**36.2**	**39.5**	**29.6**

6.5.1 Key idea

Typically, if an HMM/GMM acoustic model is adapted to an accent, the mean and the variances of all the Gaussians are modified by different methods, such as MAP or MLLR, to make the acoustic model better suitable to the accent. This kind of modification is referred to as "transformation" in this section. The idea of crosslingual accent adaptation is to use the transformation which was learned to adapt the native language to the non-native one across languages assuming that the accent stays the same. Figure 6.1 illustrates this proposed

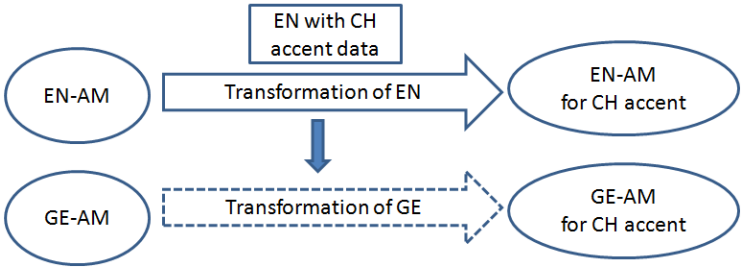


Figure 6.1: *Crosslingual accent adaptation* approach

approach for a scenario in which English and German acoustic models should be adapted to English and German with Chinese accent. In this example, the English with Chinese accent adaptation data is available but no German with Chinese accent adaptation data is provided. That means, 1) the transformation T which is used to adapt the English model to English with Chinese accent can be estimated using the provided adaptation data but 2) there is no chance to estimate the transformation to adapt the German model to German with Chinese accent. The key point is that the accent is the same, i.e. L1 stays the same and the effect of L1 on different L2 languages might share some common characteristics. Therefore, using T to adapt German models might improve the

ASR performance on German with Chinese accent. This research idea allows borrowing transformations across languages for accent adaptation if the target accent is the same.

6.5.2 Implementation using multilingual AM

Obviously, the main challenge is to determine the context dependent HMM states in the target language (e.g. in German) which should be adapted using the borrowed transformation of the source language (e.g from English). Similar states between languages are a reasonable solution. To decide which states are similar, there are several possibilities. For example, distance measures between Gaussian Mixtures, such as Kullback-Leibler distance [Kul87] can be used. Based on these distances, similar states should be adapted using the same transformation in the phonetic space. In this thesis, we propose to train a multilingual model in which the states are shared between languages (see figure 6.2). The phone set should be merged between languages if they share the same symbols in the IPA table. By doing that, the context dependent HMM states are merged together if they are similar during building the context decision tree of the multilingual acoustic model. Therefore, they are implicitly transformed by adapting the multilingual acoustic model to the accent. The main advantages of this approach are 1) that the similarity of the context dependent HMM states across languages is determined implicitly during the training and 2) that the adaptation can be performed automatically for all the languages. Furthermore, we propose to perform only MAP adaptation since in contrast to MLLR the Gaussian mixtures of each HMM state are independently adapted. This allows us to better understand the crosslingual effect in which the performance of each shared phone can be analyzed before and after applying the proposed approach.

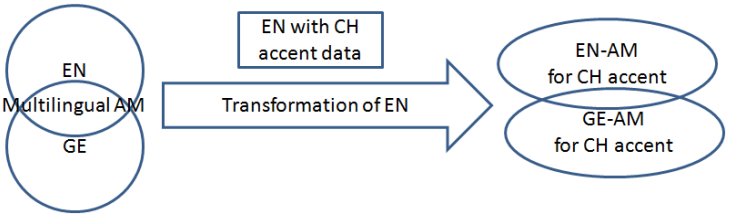


Figure 6.2: *Crosslingual accent adaptation* with multilingual AM

6.5.3 Experiments and Results

For the crosslingual accent adaptation, we conducted two experiments: The first one assumed that English with Chinese accent was not available. Therefore, we used German with Chinese accent to improve the background acoustic model. In the second experiment, German with Chinese accent was not available and therefore, English with Chinese accent was utilized for adaptation. Based on the results of the experiments in Section 6.4.1, we used not only English and German but also Mandarin data to train the multilingual model which served as the background model in both experiments. This multilingual acoustic model has 5,000 quintphones. In our case, there are 24 phones which are shared between English and German. They correspond to 1,606 context dependent states which represent 32.12% of all the states. When English quintphone states are adapted to English with Chinese accent, all the German quintphone states which are shared with English quintphone states are also adapted implicitly and vice versa. In the first experiment, when we adapted the background model on German data with Chinese accent, 2,075 states were adapted in total. Of those, 1,367 states were shared between English and German. Compared to the first experiment, less states were adapted in the second experiment. More specifically, 1,662 states were adapted using English data with Chinese accent. 1,195 of them were shared between English and German. The reason lies within the fact that the amount of German data with Chinese accent is greater than the English one. Table 6.3 summarizes the WER on English and German with Chinese accent. The results show that we achieved in total about 19.8% relative improvement on English with Chinese accent and 11.9% on German with Chinese accent without using any adaptation data of the target language compared to the monolingual baseline system. In the case of testing on English with Chinese accent, the multilingual acoustic model was adapted with German data and, therefore, more states were adapted than in the case of testing on German with Chinese accent. Therefore, it can be explained why the improvement on the English test set with Chinese accent is larger than on the German data with Chinese accent.

6.5.4 Result analysis

The results indicate that we can share data across L2 languages with the same accent to improve the ASR system on non-native speech. This can be applied to the case that we do not have any training or adaptation data of the target L2 language and the target accent. To obtain a better understanding of the ASR improvement, we performed an error analysis on phone level in which we compared the ASR errors of German and English with Chinese accent before

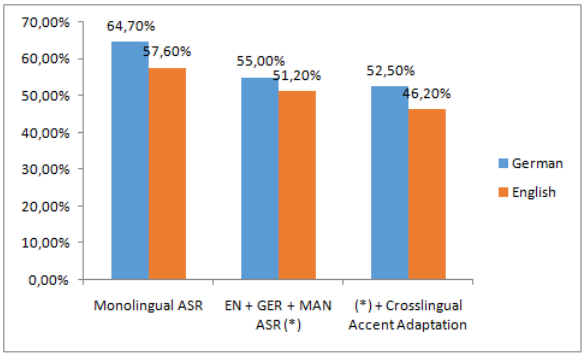


Figure 6.3: WER on German and English with Chinese accent

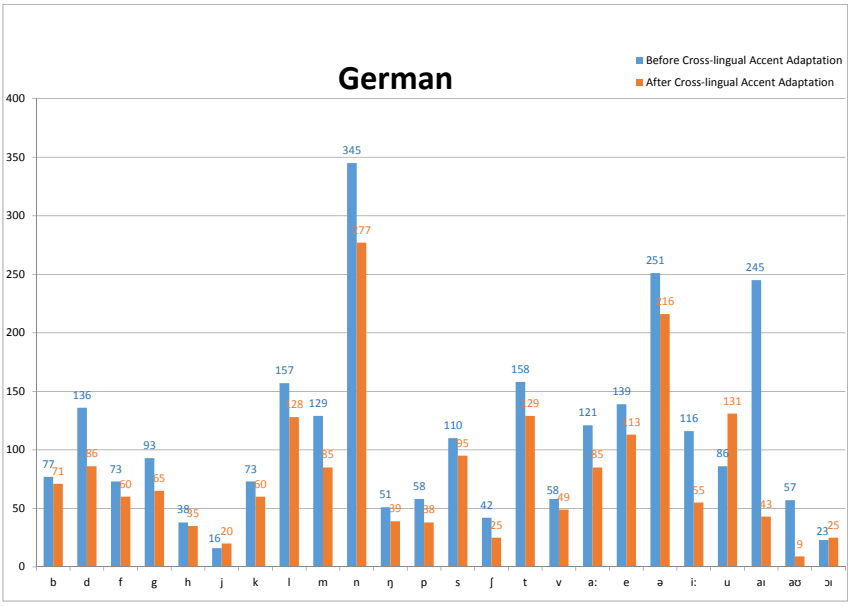


Figure 6.4: Substitution errors of shared phones before and after using *crosslingual accent adaptation* for German

and after applying *crosslingual accent adaptation*. Figures 6.4 and 6.5 show all 24 shared phones and how often they were misrecognized in the German and English test set with Chinese accent.

In total, we observed consistent improvements of these shared phones after

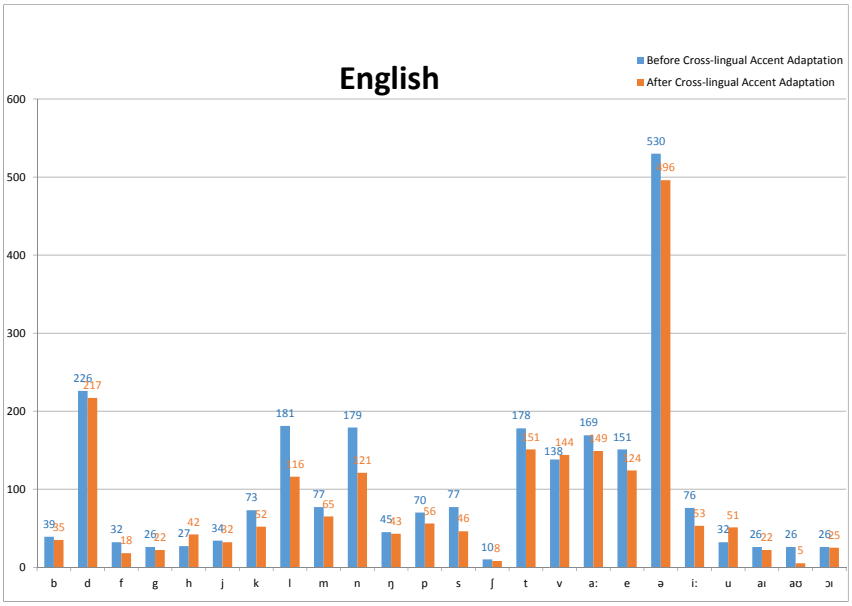


Figure 6.5: Substitution errors of shared phones before and after using *crosslingual accent adaptation* for English

applying the *crosslingual accent adaptation* approach on the German and English non-native test set. These results indicate that the L1 language has the same effect on different L2 languages, i.e. L1 native speakers may not be able to pronounce or wrongly pronounce the same phones of the L2 languages. Based on the experimental results and the error analysis, we can conclude that the improvement in our experiments is predictable. Since L1 native speakers may pronounce the same phones of L2 in the same way according to their accent, the accent transformation can be shared among different L2 languages.

6.6 Summary

This chapter presented our latest investigations of using multilingual and crosslingual information to improve automatic speech recognition performance on non-native speech. Our experimental results revealed that bilingual L1-L2 acoustic models can improve ASR performance on non-native speech. If L1 is unknown, multilingual ASR trained without L1 speech data outperforms monolingual ASR on non-native speech. For the case that no adaptation data for the tar-

get accent is available, *crosslingual accent adaptation* provided 15.8% relative improvement in average compared to the baseline system.

CHAPTER 7

Multilingual Deep Neural Network Based Acoustic Modeling For Rapid Language Adaptation

Deep neural networks (DNNs) have become state-of-art techniques for acoustic modeling in the last years. They outperform traditional Gaussian Mixture Models in various tasks with different data sets. This chapter describes an investigation on multilingual deep neural network based acoustic modeling in the context of rapid language adaptation.

7.1 Introduction

Since the late nineties, multilingual acoustic models and their use to bootstrap ASR systems for unseen languages have become one of the most important research topics in the speech community. Many interesting research works, such as [WKAM94, CC97, GG97, SW98a, Köh98, SW98b, SW98b, Köh98] were conducted in this time period. One of the most important findings was that

multilingual acoustic models outperform monolingual ones for the purpose of rapid language adaptation [SW01b].

Povey et al [PBA+10] proposed a subspace GMM framework which gives a substantial improvement over the traditional HMM/GMM. Moreover, multilingual Subspace GMM was shown to outperform the monolingual ASR system for the first time [BSA+10]. Afterwards, HMM/DNN hybrid systems that use deep neural networks (DNNs) to estimate the emission probabilities of the Hidden Markov Model (HMM) states [SLY11, DYDA12, MDH12] were successfully applied to large vocabulary ASR and led to significant improvements in various tasks with different data sets. Many recent studies [SGR12, HLY+13, HVS+13, GSR13] exploited multilingual data during DNN training in different unsupervised and supervised ways to improve the monolingual ASR performance. In these studies, it was shown that the shared hidden layer is to some extent language independent and can be used to bootstrap the DNN for a new language.

To train a multilingual acoustic model training, there are several possible ways: on a merged universal phoneset based on the international phonetic alphabet (IPA) chart, i.e. the same IPA symbols are merged across languages, or on a merged universal phoneset without merging strategies. In this thesis, we compare the two methods in the context of multilingual DNN.

Moreover, multilingual DNNs seem to work particularly well in combination with Kullback–Leibler divergence based hidden Markov modeling (KL-HMM) if only small amounts of data are available for the new language [IMGB13]. However, in [IMGB13], only small bilingual DNNs (Afrikaans and Dutch) without pre-training were evaluated.

In this thesis, we investigate the effect of IPA based phone merging on the multilingual DNN and its application to new languages. We also study multilingual DNNs in combination with KL-HMM on a large scale, involving up to five hidden layers, up to 6,000 MLP outputs and DNNs trained on up to six languages. Furthermore, we investigate how different pre-training methods influence cross-lingual DNN based acoustic modeling in the context of rapid language adaptation.

Compared to previous studies, the two main contributions of this thesis are: 1) investigating the effect of phone merging on multilingual DNNs, and 2) extensive exploration of DNN based acoustic modeling in the context of rapid language adaptation.

7.2 Related work

7.2.1 Multilingual DNN

This section summarizes the most important work on multilingual DNN acoustic modeling and its application to bootstrap AMs for new languages. [SGR12] examined the usability of unlabeled data from one or more languages to improve recognition accuracy of a different, possibly low-resource, language in a fully unsupervised fashion. They used an unsupervised RBM 2.2.2 trained with one or multiple languages to initialize the DNN of a new language. Their results showed no significant improvement between using unsupervised monolingual RBM and multilingual RBM.

The authors in [HLY+13, HVS+13] trained a multilingual acoustic model using all the multilingual training data. The softmax output layer was trained separately for each language, however the hidden layers are shared between languages. Their results showed that the shared hidden layer is language independent and can be used to bootstrap the DNN for a new language.

In contrast to other works, the authors in [GSR13] trained the multilingual MLP sequentially. They trained a network on one language and then replaced the output layer with the one corresponding to another language, borrowed the hidden layers, and fine tuned the whole network on the new language. This process was repeated for several different languages to obtain the multilingual DNN. Their results showed that the hidden layers can be shared between languages to improve accuracy.

All these works indicate that the hidden layers save some language independent information which could be learned on several languages and transferred to another language. Due to this fact, a lot of multilingual data could be used to train a large network. Then, it can be applied to bootstrap the acoustic model for a new language with only a small amount of training data.

7.2.2 KL-HMM

Recently, Imseng et al. [IMGB13] showed that multilingual DNNs work particularly well in combination with Kullback–Leibler divergence based hidden Markov modeling (KL-HMM). However, in their experiments, only small amounts of data were available for the new language. Furthermore, only DNNs with three hidden layers were used, pre-training was not applied and the setup was bilingual (Afrikaans and Dutch) rather than addressing multiple languages.

7.3 DNN training with KALDI

This section describes the key features of the KALDI DNN training recipe [ZTPK14] - part of the Kaldi ASR toolkit [PGB+11] - which we used in our study. Currently Kaldi contains two parallel implementations for DNN training. Both of these recipes support deep neural networks training which is done on top of the standard HMM/GMM training recipe. That means, the context dependent decision tree, the audio alignment and the feature transform (if it is used) are adopted from the HMM/GMM system. The neural net is trained to predict the posterior probability of each context-dependent state. During decoding, the output probabilities are divided by the prior probability of each state to form a "pseudo-likelihood" that is used in place of the state emission probabilities in the HMM [BM94].

7.3.1 First Kaldi DNN implementation

The first implementation is described in [VGBP13]. This implementation supports Restricted Boltzmann Machines (RBM) pre-training [EBC+10] - generative pre-training, stochastic gradient descent (SGD) training using NVidia graphics processing units (GPUs) and discriminative training.

7.3.2 Second Kaldi DNN implementation

The second Kaldi DNN training recipe supports parallel training on multiple CPUs. Instead of using Restricted Boltzmann Machine pre-training, the greedy layer-wise supervised training [BLPL07] or the "layer-wise backpropagation" of [SLY11] is used. A parameter defines the number of iterations in which the network should be trained before a new hidden layer is inserted between the last hidden layer and the softmax layer. This is repeated until a desired number of layers is reached.

The parallelization of the neural network training is performed on two levels: on a single machine, and also across machines. The parallelization method on a single machine involves multiple threads simultaneously updating the parameters while simply ignoring any synchronization problems. This is similar to the Hogwild! approach [NRRW11]. Furthermore, on different machines, multiple training processes are run independently using SGD on different random subsets of the data. After processing a specified amount of data, each machine writes its model to the disk. Afterwards, the averaged model parameters become the starting point for the next iteration of training.

The training recipe does also support different methods to stabilize the training, such as preconditioned SGD and enforcing the maximum change in the parameters per minibatch.

The initial and final learning rates in the training recipe need be specified by hand. During training, we decrease the initial learning rate exponentially to reach the final learning rate for a few epochs at the end. The learning rate remains unchanged during these last epochs. After the final iteration of training, the models from the last n iterations are combined via a weighted-average operation into a single model. The weights are determined via non-linear optimization of the cross-entropy on a randomly selected subset of the training data.

7.4 Multilingual DNN

For our studies, we use multilingual DNNs. We train the multilingual DNNs in two steps: 1) training on multilingual data using a universal phone set, and 2) performing cross-language model transfer by re-training the output layer on target language data. To further exploit the (limited amount of) target language data, we also perform Kullback–Leibler divergence based HMM (KL-HMM) decoding.

7.4.1 Universal phone set

To train the multilingual DNN, we investigate two different kinds of universal phone sets. The first multilingual phone set, *MUL-SEP*, is created by simply concatenating all the involved monolingual phone sets with a language identification prefix to ensure that all the phones are distinct among languages. To create the second universal phone set, *MUL-IPA*, we merge all the monolingual phones which share the same symbol in the IPA table. Obviously, the number of phones in the *MUL-SEP* phone set is larger than in the *MUL-IPA*.

To obtain the tied-state targets for the training of the multilingual DNN, we used the KALDI toolkit. More specifically, for both universal phone sets, we trained multilingual HMM/GMM systems and built multilingual decision trees to generate tied-state alignments. Furthermore in all the experiments, we used the same number of Gaussians to train the *MUL-SEP* and *MUL-IPA* acoustic models in order to provide a fair comparison between the two kinds of multilingual DNNs (Figures 7.1 and 7.2).

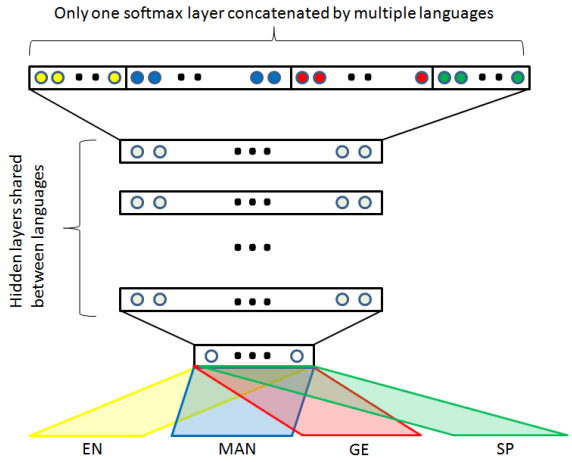


Figure 7.1: Multilingual deep neural network based on a multilingual decision tree in which the phones are not shared between languages

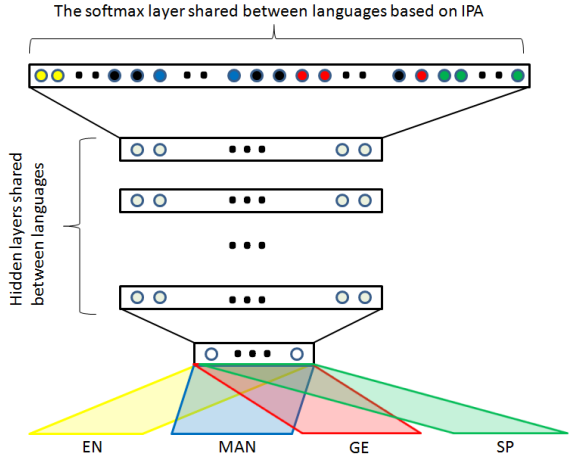


Figure 7.2: Multilingual deep neural network based on a multilingual decision tree in which the phones are shared between languages based on IPA

7.4.2 Cross-language model transfer

To bootstrap the acoustic model for a new language using multilingual DNN, the hidden layers of the multilingual DNN are shared and transferred to the new language. The multilingual softmax layer is simply replaced with a new output layer corresponding to the target language. All the weights which connect the neurons of the last hidden layer to the last layer and the biases are randomly initialized.

7.4.3 KL-HMM

In a recent study [IMGB13], it was shown that KL-HMM decoding is particularly useful if ASR systems for low-resourced languages are improved by using out-of-language data. Therefore in this thesis, we also apply KL-HMM decoding as an alternative to conventional hybrid decoding. Conventional hybrid systems directly use the MLP output to estimate the emission probability of the HMM states, hence, each HMM state only considers the output of the corresponding neuron in the softmax layer of the MLP. In contrast, the (deep) Tandem systems [SGR12] use the whole MLP output vector as speech features. However, since Tandem systems model the HMM states with Gaussian mixtures, the MLP output vector needs to be post processed and usually the dimensionality is reduced as well. The KL-HMM acoustic modeling technique can directly model high dimensional MLP output vectors. The HMM states are parametrized with reference posterior distributions (categorical distributions) that can be trained by minimizing the Kullback–Leibler divergence between the categorical distributions and the MLP output. More details about training and decoding in the KL-HMM framework can be found in, for instance, [IMBG13].

7.5 Setup

We conducted two different sets of experiments by varying the relation between the source and the target languages. Furthermore, to verify the generalization of the study, the experiments were performed with different implementations which support two state-of-the-art techniques for deep neural network training namely RBM pre-training and greedy layer-wise supervised training (see Section 7.3).

The first set of experiments is conducted with four Indo-European languages. Three source languages, namely FR, GE and SP are used to train the multilingual DNN which is then adapted to PO. Note that in this case the target language is related to the source languages.

The second set of experiments was conducted with speech data from different language families. We use EN, BG, GE and SP as representatives of Indo-European languages, MAN as a Sino-Tibetan language and JP from the Altaic language family for the multilingual DNN training. The multilingual DNN is then adapted to three different target languages CZ, HA and VN which are from three different language families. CZ and VN belong to the Indo-European and Sino-Tibetan languages, respectively. Both language families are represented in the source languages. HAU on the other hand is a language from the Afro-Asiatic language family which is not related to any of the source languages.

7.6 Results

This section presents all the experimental results of our study. Different DNNs were trained using different initialization schemes, such as random initialization (*Random-Init*), generative pre-training (*Gen-PT*) or greedy layer-wise supervised training (*GL-sup*), and served as baseline systems. Furthermore, we used different universal phone sets (described in Section 7.4 - *MUL-SEP* and *MUL-IPA*) to train the multilingual DNNs that were then used to bootstrap the monolingual DNNs, which we refer to as *DNN-MUL-SEP* and *DNN-MUL-IPA* respectively. We also performed KL-HMM decoding as an alternative to conventional hybrid decoding, referred to as *DNN-MUL-SEP + KL* and *DNN-MUL-IPA + KL*.

7.6.1 Experiments with related languages

The first set of experiments was carried out on similar languages and evaluated (only) on the Portuguese (PO) test set. All the DNNs were trained using the first DNN implementation of KALDI. We assumed to have different amounts of PO data available: the full training set (17h), and randomly selected 5h and 1h subsets. All the results are summarized in Table 7.1.

The upper part of the table shows the results without applying pre-training and the lower part shows results if the DNNs are pre-trained prior to fine-tuning. System *DNN* was pre-trained on the PO data. For all the other systems, the

Table 7.1: *Word error rates (WER) on the PO test data. The numbers in the upper part correspond to experiments without pre-training the DNNs and the numbers in the lower part to experiments with pre-training*

Amount of PO data	17 h	5 h	1 h
No DNN pre-training			
DNN (Random-Init)	21.4	25.4	34.1
DNN-MUL-SEP	20.0	23.2	29.4
DNN-MUL-SEP + KL	20.0	22.9	29.0
DNN-MUL-IPA	20.3	23.2	29.4
DNN-MUL-IPA + KL	20.0	23.1	29.0
DNN pre-training			
DNN (Gen-PT)	20.7	24.8	33.8
DNN-MUL-SEP	20.4	23.4	29.0
DNN-MUL-SEP + KL	**19.9**	23.1	28.6
DNN-MUL-IPA	20.4	23.0	29.0
DNN-MUL-IPA + KL	20.4	**22.7**	**27.8**

term with or without pre-training refers to the multilingual DNN. Afterwards, to obtain the PO DNN, the cross-language model transfer is applied.

All the DNNs used in this set of experiments had three hidden layers, each consisting of 2,000 units and were trained from 9 consecutive frames (4 preceding and 4 following frames) of 39-dimensional Mel-Frequency Cepstral Coefficients (MFCC) including deltas and double deltas. The first rows correspond to systems that only used the PO data (baselines). The Portuguese DNN was trained to estimate posterior probabilities of 2,252 tied-state triphone targets. We also evaluated cross-language model transfer by bootstrapping the DNNs with hidden layers trained on FR, GE and SP data, using *MUL-SEP* and *MUL-IPA* phone sets. The *MUL-SEP*-DNN and the *MUL-IPA*-DNN were trained to estimate posterior probabilities of 3,338 and 3,139 tied-state targets, respectively, obtained from the multilingual decision trees. Note that, for each type of multilingual DNNs, we trained two different networks, using random initialization and generative RBM pre-training. We also evaluated KL-HMM based decoding for each scenario. For the experiments on the whole PO training set, we fixed the number of KL-HMM states to 20,000. For the subsets of 5 h and 1 h, we used 10,000 and 6,000 KL-HMM states, respectively.

Table 7.1 reveals the following trends: The cross-language model transfer based on multilingual DNN (*DNN-MUL-SEP* and *DNN-MUL-IPA*) consistently outperforms the PO baseline system (DNN), trained with random initialization or generative RBM pre-training. However, it is not clear whether using genera-

tive RBM pre-training to train the multilingual DNN helps to improve the ASR performance on the target language.

Moreover, using KL-HMM, the performance is the same or better. The ASR performance tends to improve more in case of small amounts of training data while only marginal performance differences are observed if the whole PO training set is used. In combination with multilingual DNN which was pre-trained with RBM, we obtained the best WER on the PO test set. The difference between the two different universal phone sets seems to be rather small, but in the case of less training data, using IPA seems to be beneficial.

7.6.2 Experiments with non-related languages

Multilingual DNN

In the second set of experiments, we used the second DNN implementation of KALDI to train two different multilingual DNN AMs with *MUL-SEP* and *MUL-IPA* phone set using the training data of six different languages (BG, EN, GE, JA, MAN, and SP). We apply the greedy layer-wise supervised training to train the multilingual DNN (DNN GL-sup). MFCC features with the first 13 coefficients concatenated with 5 left and 5 right neighbors were used directly as input of the DNN after fMLLR transformation. For each multilingual DNN, 6,000 tied-state triphones were trained. The DNN had 5 hidden layers, each consisting of 1,500 units. We also applied crosslingual model transfer[1] re-trained the DNN for each target language. Table 7.2 shows the results. Crosslingual model transfer consistently improved WER compared to the greedy layer-wise supervised training and fine-tuned DNN that used the monolingual data only. The *DNN-MUL-IPA* systems yielded slightly better performance than the *DNN-MUL-SEP* systems in the case of Bulgarian, English and Japanese. For German, Mandarin and Spanish, the WER is the same.

Rapid language adaptation to new languages

For language adaptation experiments, we conducted two different experiments on the Czech, Hausa and Vietnamese GlobalPhone data set: with the full amount of training data and with only small amount of training data. Based on the result of the first set of experiments from 7.6.1, we applied KL-HMM based

[1]Note that in this context, the target language was already part of the multilingual DNN training, hence the term crosslingual model transfer may be misleading. However, the re-training procedure is as described in Section 7.4.

Table 7.2: *Word error rates (WER) on BG, EN, GE, JA, MAN, and SP test data using greedy layer-wised supervised training DNN and DNNs which were pre-trained using multilingual DNNs*

Systems	BG	EN	GE	JA	MAN	SP
DNN (GL-sup)	17.4	9.9	6.2	16.8	12.3	14.9
DNN-MUL-SEP	16.8	9.5	**5.8**	16.2	**11.8**	**14.3**
DNN-MUL-IPA	**16.7**	**9.2**	**5.8**	**16.1**	**11.8**	**14.3**

decoding only with small amounts of training data. First, we used all the training data and trained the DNN for Czech, Hausa and Vietnamese. Table 7.3 summarizes the WER on CZ, HA and VN test data. Again, the crosslingual model transfer yielded consistent improvements compared to the baseline system which was greedy layer-wise supervised trained and fine-tuned only with monolingual data of the target language.

In this set of experiments, using IPA to merge the phone set of the multilingual DNN seems to slightly improve the ASR system in the case of CZ and HA. However, the syllable ER increases a bit in the case of Vietnamese. Note that, in the case of Hausa, even though the target language and the source languages are completely unrelated, we observed up to 6% relative improvement.

Table 7.3: *ASR performance on CZ, HA, and VN test data trained with full amount of training data*

Systems	CZ	HA	VN
DNN (GL-sup)	9.9	10.1	10.0
DNN-MUL-SEP	9.3	9.8	**8.6**
DNN-MUL-IPA	**9.2**	**9.5**	8.8

Second, we assume that only a small amount of training data - one hour - for each target language is available. The results in Table 7.4 show that by using multilingual DNN, we observed larger improvements over the baseline system than in the previous experiment. This indicates that the multilingual DNN is very useful if the amount of training data is rather small. The *DNN-MUL-IPA* is slightly better than the *DNN-MUL-SEP* system in the case of Hausa. In the case of Czech and Vietnamese, the ASR performance is only marginally different. However, if we use KL-HMM based decoding, we consistently obtained better ASR performance by using the *DNN-MUL-SEP*.

Table 7.4: *ASR performance on Czech, Hausa and Vietnamese test data trained with one hour of training data*

Languages	CZ	HA	VN
DNN (GL-sup)	16.9	16.1	32.1
DNN-MUL-SEP	14.0	13.6	27.1
DNN-MUL-SEP + KL	**13.1**	**12.0**	**26.6**
DNN-MUL-IPA	13.9	13.3	27.0
DNN-MUL-IPA + KL	13.4	12.3	26.8

7.7 Summary

This chapter presented an extensive investigation of multilingual DNN based acoustic modeling in the context of rapid language adaptation. On different languages, we found that Kullback–Leibler divergence based hidden Markov models in combination with crosslingual model transfer yields the best performance. The performance improvement is more pronounced in low-resource scenarios. Table 7.5 summarizes the relative improvement of using crosslingual model transfer based on multilingual DNN in combination with KL-HMM over the baseline DNN system. Moreover, our experiments also suggest that it is not

Table 7.5: *Relative improvement of using crosslingual model transfer based on multilingual DNN in combination with KL-HMM in low-resource scenarios*

Language	CZ	HA	PO	VN
Relative improvement (%)	22.5	25.4	17.8	17.1

necessary to manually derive IPA based universal phone sets for multilingual DNN training.

CHAPTER 8

Multilingual Language Model For Code-Switching Speech

Code-Switching speech is a common phenomenon in multilingual communities which becomes more popular due to the globalization effect. This chapter describes the investigation of language modeling for Code-Switching speech. The idea is to analyze textual features which might have potential to predict Code-Switches and afterwards, integrate those features into state-of-the-art language models such as recurrent neural network language models (RNNLM) and factored language models (FLM) for the Code-Switching task. Finally, an investigation of Code-Switching attitudes is presented.

8.1 Introduction

Code-Switching speech is defined as speech that contains more than one language ('code'). The switch between languages may happen between or within an utterance. It is a common phenomenon in many multilingual communities where people of different cultures and language background communicate with each other [Aue99a]. For the automated processing of spoken communication in these scenarios, a speech recognition system must be able to handle Code-Switches. In general, there are two possible ways to build an automatic

speech recognition system for Code-Switching speech. In the first approach, a language identification system is used to split the Code-Switching speech into different monolingual parts and, afterwards, monolingual recognizers are applied to the corresponding speech segments. This method is rather straightforward since the monolingual systems may be already available. However, we lose semantic information between the segments and the mistakes of the language identification system cannot be recovered. Especially for short speech segments (e.g. shorter than 3 seconds), the language identification system performance is not reliable. The second approach applies an integrated system with multilingual models (acoustic model, dictionary and language model). Compared to the first approach, the semantic information can be used between languages. However, there might not be enough bilingual training data. This is a challenge for the integrated system. While there have been promising research results in the area of acoustic modeling, only few approaches so far address Code-Switching in the language model. Due to the lack of Code-Switching text data, language modeling is a challenging task. Traditional n-gram approaches may not provide reliable estimates. Hence, more general features than words should be integrated into the language models.
Recurrent neural networks and factored language models provide the possibility to add different features to each word. Additionally, it has been shown that recurrent neural network language models (RNNLMs) improve perplexities and error rates in speech recognition systems in comparison to traditional n-gram approaches [MKB$^+$10, MKB$^+$11, YPC12]. One reason for that is their ability to handle longer contexts. On the other hand, factored language models (FLMs) have been used successfully for languages with rich morphology due to their ability to process syntactical features, such as word stems or part-of-speech tags (POS) [BK03, EDSN10].
In this chapter, we describe our approach to develop a multilingual language model for the Code-Switching task. We apply recurrent neural network language models and factored language models in which features, such as POS tags or language identifiers (LID) are integrated to improve the LM performance. Furthermore, a comparison between the models and a detailed analysis are provided to explain the results. Additionally, we show that the linear interpolation of RNNLM and FLM provides the best performance on the SEAME corpus. Figure 8.1 illustrates our Code-Switching system.
Finally, we show that clustering speakers according to their Code-Switching attitudes leads to improvements in terms of perplexity for each test speaker. These improvements also transform into error rate reductions.

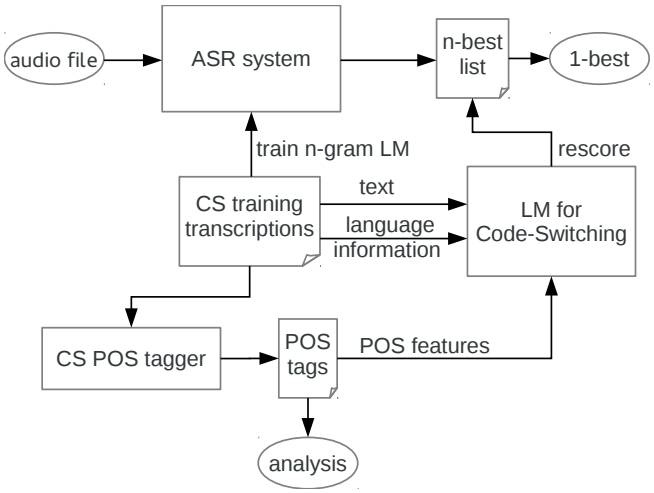


Figure 8.1: Overview: our Code-Switching system

8.2 Related Work

Linguistic analyses of the Code-Switching phenomenon help to better understand the task and challenges and, thus, might help to create an appropriate language model. Hence, various studies on Code-Switching are described. Furthermore, recent developments in the research on modeling Code-Switching speech are summarized. Finally, we also give a short overview of using recurrent neural network language models and factored language models in other contexts.

8.2.1 The Code-Switching phenomenon

Linguistic analyses of Code-Switching were already performed in the 1980s and 1990s. Researches mainly covered Spanish-English-Code-Switching in Puerto Rican communities in the United States [Pop78, Pop80] and Italian-German or Italian-English-Code-Switching [Aue99b]. This subsection highlights the most important works and results.
[Pop78] observes different Code-Switching types. The author finds that language changes may occur in different contexts, such as between full sentences, between conjoined sentences, at interjections, between major noun and verb phrases, between verb and object noun phrases, etc. However, the first few types which integrate more words of the second language are detected more

often than the other types. Furthermore, two linguistic constraints for Code-Switching are described: The free morpheme constraint and the equivalence constraint. The author disagrees with statements that Code-Switching occurs at random points and states that it is rule-governed. Hence, Code-Switching points may be predictable. Finally, the paper concludes that people speak that language which they feel most comfortable with. Hence, the greater the knowledge of the speakers, the higher is the integration of a different language into their mother tongue. Especially with less educated people, pauses, hesitations or repair mechanisms, such as false starts, appear before language changes.
In a second paper, [Pop80] analyzes the speech of 20 Puerto Rican residents in the United States. The results fit to the observations and statements of the first paper: A fluent bilingual may switch the language at different syntactic points, even intra-sentential, without pauses or hesitations, while a non-fluent favors switches at sentence boundaries and usually pauses before the switch. Nevertheless, both speaker groups do not violate the free morpheme or the equivalence constraint. Furthermore, the combination of the two languages does not violate grammatical structures. One of the main contributions of the paper is an analysis of extra-linguistic factors that may effect Code-Switching behaviors. Especially, the factors gender, age of second language acquisition, bilingual ability and work place show a statistical significance at a level of 0.001 and can be regarded as independent of other factors. It is, for example, discovered that women significantly favor intra-sentential Code-Switching while men prefer extra-sentential switches.

8.2.2 Modeling Code-Switching speech

The authors of [SL08a] applied different machine learning algorithms (for instance the Naive Bayes Classifier or Value Feature Interval) trained on textual features to predict Code-Switching points. As features, they use word form, language ID, part-of-speech tags and the position of the word relative to the phrase. The work uses a Spanish-English-Code-Switching corpus containing 40 minutes of conversational speech. However, Spanish and English are not equally distributed in the corpus. In fact, English is the predominant language. The authors detect that their machine learning algorithms perform better than Support Vector Machines, C4.5 decision trees and neural networks on their task. As evaluation measures, they use precision, recall and F-measure. Furthermore, they artificially generate Code-Switching sentences and ask people who are familiar with Code-Switching to evaluate their naturalness.
[CCLC06] develop a large vocabulary speech recognition system for Cantonese-English Code-Switching speech which is common in Hong-Kong. The authors describe two different approaches to a Code-Switching recognition system: The first approach involves language boundary detection (using language

specific phonological and acoustic properties) and a monolingual recognition afterwards. The second approach uses a cross-lingual ASR system. In their work, the authors develop a two-pass decoding algorithm. In the first pass, a syllable/word lattice is generated using a cross-lingual acoustic model and a bilingual dictionary. Then, a syllable-to-character dictionary is applied to generate a character graph. Furthermore, language boundaries are detected. In the second pass, the Chinese character sequence is decoded using a language model that is based on Cantonese characters (trigram) and a small number of English word classes. To find an appropriate language model, four different n-gram models are trained and compared: The first one is a monolingual model which regards all the foreign words as out-of-vocabulary words (OOV). The second model provides all the foreign words with the same probability. The third one gives the foreign words the probability of their translated equivalent. The last model is class-based and clusters all foreign words into their part-of-speech classes. The language models are evaluated in a phonetic-to-text conversion task. The class-based language model performs better than the other language models. The authors assume that the reason may be training data sparseness.

8.2.3 Recurrent neural networks language models

In the last years, neural networks have been used for a variety of tasks. [MKB+10] introduced a refined form of neural networks for the task of language modeling. The so-called recurrent neural networks are able to handle long-term contexts since the input vector does not only contain the current word but also the previous values of the neurons of the hidden layer. It is shown that these networks outperform traditional language models, such as n-grams which only contain very limited histories. In [MKB+11], the network is extended by factorizing the output layer into classes to accelerate the training and testing processes.
Recently, further information has been added to the recurrent neural network. Shi et al. [YPC12] augment the input layer to model features, such as topic information or part-of-speech tags.
Furthermore, language model adaptation has been investigated, such as in [KMKB11]. The authors show that adaptation of recurrent neural network language models in form of one-iteration re-training on the hypothesis leads to improvements in terms of word error rate if the adapted models are applied for rescoring.

8.2.4 Factored language models

A factored language model refers to a word as a vector of features (factors), such as the word itself, morphological classes, part-of-speech tags or word stems. Hence, it provides a possibility to integrate syntactical features into the language modeling process. [BK03] show that factored language models are able to outperform standard n-gram techniques in terms of perplexity. In the same paper, generalized parallel back-off is introduced. This technique can be used to generalize traditional backoff methods and to improve the performance of factored language models. Due to the integration of various features, it is possible to handle rich morphology in languages like Arabic or Turkish [DK04, EDSN10].

8.3 Linguistic Analysis

8.3.1 Description of the data corpus

SEAME (South East Asia Mandarin-English) is a conversational Mandarin-English Code-Switching speech corpus. It has been recorded from Singaporean and Malaysian speakers by [LTCL10]. It was used for the research project 'Code-Switch' jointly performed by Nanyang Technological University (NTU) and Karlsruhe Institute of Technology (KIT). The recordings consist of spontanously spoken interviews and conversations of about 63 hours of audio data. For the language modeling task, all hesitations are deleted and the transcribed words are divided into four categories: English words, Mandarin words, particles (Singaporean and Malaysian discourse particles) and others (other languages). The average number of language changes between Mandarin and English is 2.6 per utterance. The duration of monolingual segments is very short: More than 82% English and 73% Mandarin segments last less than 1 second, while the average duration of English and Mandarin segments is only 0.67 seconds and 0.81 seconds respectively. In total, the corpus contains 9,210 unique English and 7,471 unique Mandarin vocabulary words. It is divided into three disjoint sets (training, development and test set). Table 8.1 lists the statistics of the SEAME corpus in these sets.

8.3.2 Prediction of Code-Switching points

Similar to the investigations summarized in Section 8.2.1, we perform an analysis of textual features that trigger language changes in the SEAME data corpus. We concentrate on words and part-of-speech tags because an analysis

Table 8.1: *Statistics of the SEAME corpus*

	Train set	Dev set	Eval set
# Speakers	139	8	8
# Utterances	48,040	1,943	1,018
# Token	525,168	23,776	11,294

in [Bur10] showed that those are the most important trigger events. We rank them according to their Code-Switching rate. The Code-Switching rate for each word or part-of-speech tag is calculated by the number of occurrences of the word or tag in front of a Code-Switching point divided by the total number of occurences in the entire text. In our analysis, we consider only those words which appear more than 1,000 times in the text, corresponding to more than 0.2% of the entire word tokens.

Trigger words

We analyze which words occur frequently immediately in front of Code-Switching points. Table 8.2 shows the top five Mandarin and the top five English words preceding a Code-Switching point.

Table 8.2: *Mandarin and English trigger words for Code-Switching points*

word	frequency	CS-rate
那个(that)	5261	53.43 %
我的(my)	1236	52.35 %
那些(those)	1329	49.44 %
一个(a)	2524	49.05 %
他的(his)	1024	47.75 %
then	6183	56.25 %
think	1103	37.62 %
but	2211	36.23 %
so	2218	35.80 %
okay	1044	34.87 %

Part-of-speech tags as trigger

Part-of-speech tagger To be able to assign part-of-speech tags to our bilingual text corpus, we apply the POS tagger described in [Bur10]. It consists of

two different monolingual (Stanford log-linear) taggers [TKMS03, TM00] and a combination of their results. While [SL08b] pass the whole Code-Switching text to both monolingual taggers and combine their results using different heuristics, in this work, the text is splitted into different languages first. The tagging process is illustrated in figure 8.2 and further described in the following.

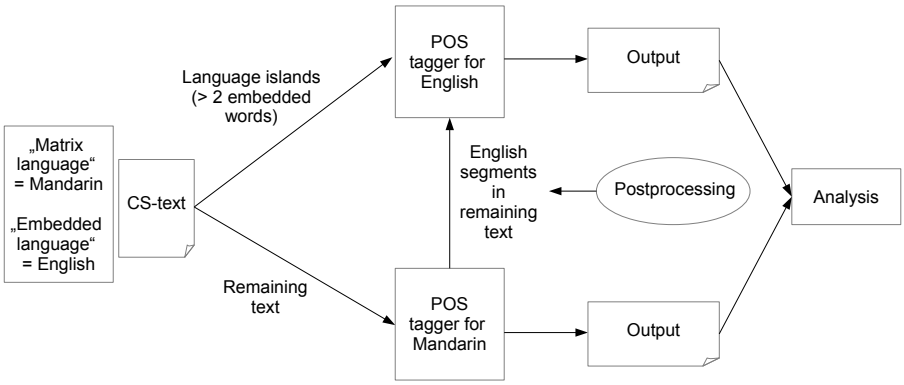


Figure 8.2: Part-of-speech tagging of Code-Switching speech

First, Mandarin is determined as the matrix language (the main language of an utterance) and English as the embedded language. If three or more words of the embedded language are detected, they are passed to the English tagger. The rest of the text is passed to the Mandarin tagger, even if it contains foreign words. The idea is to provide the tagger as much context as possible. Since most English words in the Mandarin segments are falsely tagged as nouns by the Mandarin tagger, we extend the original approach of [Bur10] with a post-processing step. We pass all the foreign words of the Mandarin segments to the English tagger in order to replace the wrong tags with the correct ones.

POS trigger Analysis After having tagged the Code-Switching text, we select those tags that possibly predict Code-Switching points. The results are shown in table 8.3. First, we consider only those tags that appear in front of a Code-Switching point from Mandarin to English. Second, we investigate the tags preceding a Code-Switching point from English to Mandarin. In each case, only those tags are counted that occur more than 250 times in the text. It can be detected that Code-Switching points are most often triggered by determiners in Mandarin and by nouns in English. This seems reasonable since it is possible that a Mandarin speaker switches for the noun to English and immediately afterwards back to Mandarin. It also corresponds to previous investigations as described in section 8.2.

Table 8.3: *Mandarin and English POS that trigger Code-Switching points*

Tag	meaning	frequency	CS-rate
DT	determiner	11276	40.44%
DEG	associative 的	4395	36.91%
MSP	other particle	507	32-74%
VC	是	6183	25.85%
DEC	的 in a relative-clause	5763	23.86%
NN	noun	49060	49.07%
NNS	noun (plural)	4613	40.82%
RP	particle	330	36.06%
RB	adverb	21096	31.84%
JJ	adjective	10856	26.48%

8.4 Language Modeling of Code-Switching Speech

This section describes our Code-Switching language models. We integrate more general features than words into recurrent neural networks and factored language models. As features, we use part-of-speech tags and language identifiers.

8.4.1 Extension of the recurrent neural network language model for Code-Switching speech

Figure 8.4.1 illustrates the recurrent neural network language model for Code-Switching speech. Two main extensions of this work are the integration of features, such as POS tags into the input layer and the factorization of the output layer using language information.
Vector $w(t)$ forms the input of the recurrent neural network. It represents the current word using 1-of-N coding. Thus, its dimension equals the size of the vocabulary. Vector $s(t)$ contains the state of the network. It is called 'hidden layer'. The network is trained using back-propagation through time (BPTT), an extension of the back-propagation algorithm for recurrent neural networks. With BPTT, the error is propagated through recurrent connections back in time for a specific number of time steps t. Hence, the network is able to remember information for several time steps. The matrices U, V and W contain the weights for the connections between the layers. These weights are learned during the training phase. In the work of [MKB$^+$11], the output layer is factorized into classes to accelerate the training and testing processes. Every word belongs to exactly one class. The classes are formed during the training phase depending on the frequencies of the words. Vector $c(t)$ contains the probabilities for each class and vector $y(t)$ provides the probabilities for each word given its class.

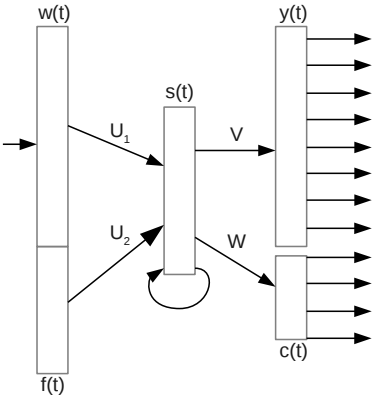


Figure 8.3: RNNLM for Code-Switching
(based upon a figure in [MKB+11])

Hence, the probability $P(w_i|history)$ is computed as shown in equation 8.1.

$$P(w_i|history) = P(c_i|s(t))P(w_i|c_i, s(t)) \tag{8.1}$$

In our extension, the classes of the output layer do not depend on word frequencies but on languages. We use the language categorization described in section 8.3.1. Therefore, our model consists of four classes: One class for all English words, one for all Mandarin words, one for other languages and one for particles. This corresponds to the Code-Switching task because, first, the probability of the next language is computed and, second, the probability of each word given the language. Furthermore, we extend the input layer by concatenating vector $w(t)$ with vector $f(t)$ which provides features corresponding to the current word. According to the analysis described in section 8.3.2, we use POS tags as features. We do not use trigger words as feature input for the network because they are implicitly modeled by vector $w(t)$. Vector $f(t)$ consists of 67 elements since the Mandarin words in the vocabulary of the SEAME transcriptions are assigned to 31 POS tags and the English words to 34 POS tags. In addition, the words classified as other languages and the particles form own classes. For each word, a relationship to its POS tag is established. Thus, during the training and testing phases, not only the current word is activated but also its feature. Because the POS tags are integrated into the input layer, they are also propagated into the hidden layer and back-propagated into its history $s(t)$. Thus, not only the previous feature is stored in the history but also all features several time steps in the past. In equation 8.1, the term $P(c_i|s(t))$ computes the next language c_i using not only information about previous words, but also about previous features.

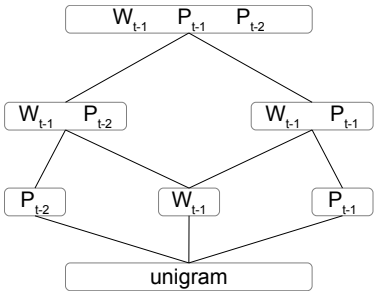


Figure 8.4: Backoff graph of the FLM

8.4.2 Integration of POS and LID into factored language models

Factored language models (FLM) are another approach to integrate syntactical features into the language modeling process, such as part-of-speech tags or language identifiers. Each word is regarded as a sequence of factors which are used for the computation of n-gram probabilities. If a sequence of factors has not been detected in the training data, backoff techniques will be applied. Since the number of possible parameters and backoff paths is rather high, a genetic algorithm as described by [DK04] is applied. [DK04] show that this method outperforms both knowledge-based and randomized choices. For our task of Code-Switching, we develop two different models: One model with part-of-speech tags as features and one including also language information tags. Figure 8.4 illustrates the backoff graph which has been obtained by the genetic algorithm for the case of part-of-speech tags as features. The different paths are combined by averaging their probability results. The model uses the last word and the two previous POS tags to calculate the probability of the next word.

8.4.3 Experimental results

LM performance

Baseline A traditional 3-gram language model will serve as baseline model in the following experiments. It is trained on the Code-Switching transcriptions using the SRI language modeling toolkit [Sto02].

RNNLM The first recurrent neural network which we build is a standard model without additional input features or classes. It is also trained using the Code-Switching transcriptions. The size of the hidden layer is set to 50 and the BPTT algorithm runs in a block mode with a block size of ten for five steps. These parameters have been tuned on the development set. The model has a perplexity of 246.60 on the development set and 287.88 on the evaluation set.

RNNLM (LID) Second, a recurrent neural network with a factorized output layer is developed. All the other parameters stay the same as in the baseline system to ensure comparability. For the output layer, we use language classes as described in 8.4.1. This approach achieves a perplexity of 239.64 on the development set and 269.71 on the evaluation set. Hence, the computation of the words depending on their languages improves the performance of the language model in terms of perplexity.

RNNLM (POS) In another experiment, the input layer of the RNNLM is extended with POS tags. This achieves a perplexity of 233.50 on the development set and 268.05 on the evaluation set. Apparently, the *RNNLM (POS)* system outperforms the *RNNLM (LID)* system.

RNNLM (POS+LID) Finally, a network is generated with a combination of both techniques. It outperforms all previous models. The perplexity of this language model is 219.85 and 239.21 on the development set and evaluation set, respectively.
The results indicate that the integration of POS features or LID improves the RNNLM in Code-Switching task. Combining both features performs the best on the development set and evaluation set.

FLM (POS) Beside RNNMLs, we also train factored language models. Our first FLM uses words and part-of-speech tags as factors. Its backoff graph has been illustrated in figure 8.4. This model has a perplexity of 260.05 and 269.15 on the development set and the evaluation set, respectively.

FLM (POS+LID) For the second factored language model, we also add LID information into the factor set. By doing so, the performance is improved to 256.78 on the development set and to 265.25 on the evaluation set. However, the improvement is rather small compared to the results in the experiments with the RNNLM.

Table 8.4 summarizes the results of our different models on the development and test set. It can be noticed that both the RNNLM and the FLM models perform better than the traditional 3-gram model. Hence, adding syntactical features and language identification improves the word prediction. However, the RNNLM outperforms the FLM.

Table 8.4: *Perplexity results*

Model	dev set	test set
Baseline 3-gram	285.87	285.25
RNNLM	246.60	287.88
RNNLM (LID)	239.64	269.71
RNNLM (POS)	233.50	268.05
RNNLM (POS + LID)	**219.85**	**239.21**
FLM (POS)	260.05	269.15
FLM (POS + LID)	256.78	265.25

Backoff level analysis

To understand the different results of the RNNLM and the FLM, a backoff level analysis similar to the one described in [OSNG12] is performed. For each word, the backoff level of the n-gram model is observed. Then, a level-dependent perplexity is computed for each model as shown in equation 8.2.

$$PPL_k = 10^{-\frac{1}{N_k} \sum_{w_k} log_{10} P(w_k|h_k)} \tag{8.2}$$

In the equation, k denotes the backoff-level, N_k the number of words on this level, w_k the current word and h_k its history. Table 8.5 shows the number of occurrences of each backoff-level and the level-dependent perplexities of each model on the development set.

Table 8.5: *Backoff-level-dependent PPLs*

	1-gram	2-gram	3-gram
# occurences	6894	11628	6226
Baseline 3-gram	5,786.24	165.82	28.28
FLM (pos)	4,950.31	**147.70**	30.99
RNNLM	**3,231.02**	151.67	**21.24**

In the case of backoff to the 2-gram, the FLM provides the best perplexity while for the 3-gram and backoff to the 1-gram, the RNNLM performs best. This may

be correlated to the better over-all perplexity of the RNNLM in comparison to the FLM. Nevertheless, backoff to the 2-gram is used about twice as often as backoff to the 1-gram or the 3-gram.

8.4.4 Language model interpolation

The different results of RNNLM and FLM show that they provide different estimations of the next word. Thus, a combination of them may reduce the perplexities of table 8.4. Hence, we apply linear interpolation to the probabilities of each two models as shown in equation 8.3.

$$P(word) = \lambda \cdot P_{M1}(word) + (1 - \lambda) \cdot P_{M2}(word) \tag{8.3}$$

P_{M1} denotes the probability provided by the first model and P_{M2} the probability from the second model. Table 8.6 shows the results of this experiment. The weights are optimized on the development set. The interpolation of RNNLM and FLM leads to the best results. This may be caused by the superior backoff-level-dependent PPLs in comparison to the 3-gram model.

Table 8.6: *Perplexities after interpolation*

Model	weight	PPL on dev	PPL on eval
FLM + 3-gram	0.7, 0.3	211.13	227.57
RNNLM + 3-gram	0.8, 0.2	206.49	227.08
RNNLM + FLM	0.6, 0.4	**177.79**	**192.08**

8.5 Code-Switching Attitude Dependent Language Modeling

8.5.1 Speaker dependent analysis

The analysis described in section 8.3.2 shows Code-Switching rates up to less than 50%. Thus, predictions based on these probabilities might not be reliable. The reason could be that one speaker switches very often after a specific tag while other speakers do not. Hence, a speaker dependent analysis should be performed. The Code-Switching rate for each tag is computed for each speaker. Then, minimal, maximal, mean values and standard deviations are calculated. Indeed, the spread between minimal and maximal values is very high for most

of the tags. Figure 8.5 shows this distribution of the speaker dependent Code-Switching rates for all the tags that appear more than 250 times in the text.

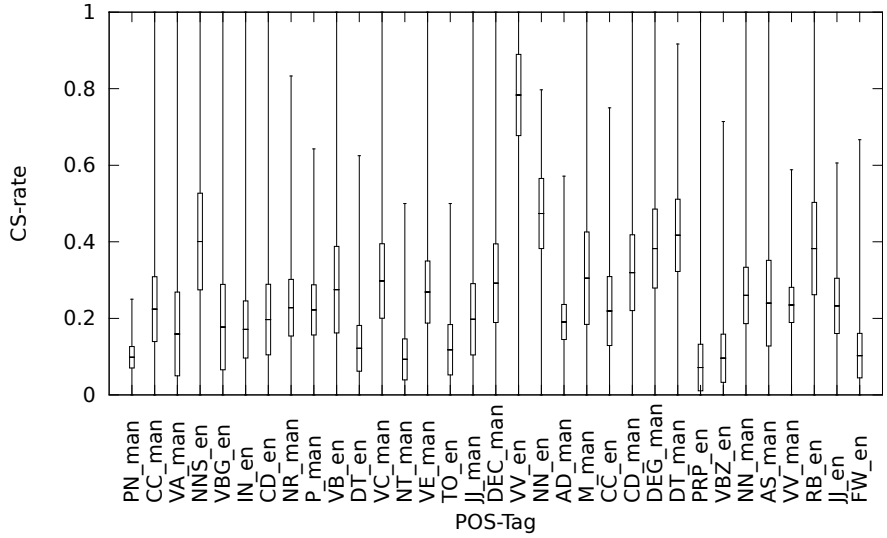


Figure 8.5: Distribution of speaker dependent Code-Switching rates

To sum up, whether a part-of-speech tag triggers a Code-Switching event seems to be speaker dependent. This corresponds to previous investigations as described in section 8.2. Hence, a model that combines all the individual deviations cannot be very precise.

8.5.2 Clustering speakers according to their Code-Switching attitude

As shown in the previous section 8.5.1, Code-Switching attitudes may be speaker dependent. However, there might be speakers who share similar habits in their usage of languages. Hence, we cluster the manual transcriptions of all the speakers of our training data into K different groups to describe different Code-Switching attitudes. After that, we are able to adapt our language model to those classes. Thus, we obtain K different language models that model Code-Switching more precisely and, therefore, achieve better recognition results.

Text Clustering We apply the k-means algorithm to cluster the training transcriptions. As similarity measure, we choose the cosine similarity because it

was successfully applied to cluster documents in the past. The following equation shows the computation of the cosine similarity $Sim(d_1, d_2)$ and the calculation of a distance measure $Dist(d_1, d_2)$ based on this. d_1 denotes a vector representing the Code-Switching attitude of speaker 1 and d_2 a vector for speaker 2.

$$\begin{aligned} Sim(d_1, d_2) &= (d_1.d_2)/(||d_1|| \cdot ||d_2||) \\ Dist(d_1, d_2) &= 1 - Sim(d_1, d_2) \end{aligned} \tag{8.4}$$

For the Code-Switching attitude modeling, we define the vectors d as follows:

$$d = [f_{cs}(POS_1)/f(POS_1), ..., f_{cs}(POS_n)/f(POS_n)] \tag{8.5}$$

$f_{cs}(POS_i)$ denotes the number of switches after part-of-speech tag i of the given speaker while $f(POS_i)$ refers to the number of all occurences of the tag. Hence, the vector is a collection of speaker-dependent Code-Switching rates.
The most important parameter in the clustering process is the cluster size. Hence, different sizes are tested and evaluated based on the perplexities of adapted RNNLMs on the development set. Using the data of the speakers in each cluster, we perform one-iteration re-training of the RNNLM to obtain one Code-Switching attitude dependent language model per cluster. Table 8.7 shows the minimum and maximum perplexity for the eight development set speakers in order to detect the most appropriate cluster size.

Table 8.7: *Minimum and maximum perplexity on the development set*

Speaker	Baseline	2 classes	3 classes	4 classes	5 classes
Spk 1	257.5	234.3 - 270.6	234.1 - 270.6	233.4 - 267.6	237.3 - 275.0
Spk 2	221.0	194.8 - 219.0	194.7 - 219.0	194.4 - 216.5	197.9 - 222.2
Spk 3	253.3	242.9 - 283.2	243.5 - 283.4	242.9 - 280.3	242.0 - 289.0
Spk 4	201.3	186.1 - 213.4	186.7 - 213.6	186.0 - 212.3	188.4 - 217.1
Spk 5	339.5	299.7 - 355.3	299.8 - 355.8	299.6 - 349.8	303.2 - 367.0
Spk 6	151.9	135.0 - 156.8	135.1 - 156.8	134.9 - 156.7	135.5 - 160.8
Spk 7	225.8	222.0 - 251.8	222.0 - 250.7	223.6 - 252.7	220.5 - 279.6
Spk 8	194.4	189.3 - 207.0	189.3 - 206.3	189.0 - 207.6	191.1 - 222.7

It can be noted that the results of two, three and four classes are quite similar and superior to a cluster size of five. Nevertheless in all cases, there are classes which lead to an improvement of the perplexity in comparison to the baseline model. Although the worst result per cluster performs worse than the baseline, most of the classes of each cluster lead to an improvement. These results support the speaker dependent analysis: It is possible to adapt the language model to individual Code-Switching attitudes.
The three best cluster sizes (2 classes, 3 classes and 4 classes) are further evaluated regarding their word error rate reduction in the rescoring process. The

experiment results in a best cluster size of 3 classes. This seems to be reasonable since two classes might not cover enough different speaker attitudes and four or more classes might not contain enough training data per class. Hence, a cluster size of three is chosen for further evaluations.

Analysis The following figures 8.6, 8.7 and 8.8 show for the example of three classes that the clustering process has helped to decrease the spread of the Code-Switching attitudes. There are still tags for which the clustered speakers show different attitudes but there are also tags for which their attitude is quite similar. For example, the spread of the English tag 'NN' (noun) is discriminated into upper and lower values by the classes.
Further analyses show that, on the one hand, the classes divide different nationalities while, on the other hand, the gender of the speakers or the speaking style is similar in all the classes. Hence, Code-Switching attitude seems to be dependent on the nationality but not on the gender or style. Table 8.8 summarizes those results for the example of three classes.

Table 8.8: *Analysis of the speakers that are clustered into one class*
(f: female, m: male, conv: conversation, interv.: interview)

Class	nationalities	gender	style
1	66 % Malaysia, 34 % Singapour	58 % f., 52 % m.	5 % conv., 95 % interv.
2	7 % Malaysia, 93 % Singapour	55 % f., 45 % m.	47 % conv., 53 % interv.
3	0 % Malaysia, 100 % Singapour	66 % f., 34 % m.	29 % conv., 71 % interv.

8.5.3 Adapted language modeling

Our clustering process results in a division of the SEAME transcriptions into three different texts (one text for each Code-Switching attitude). With these data, we adapt the n-gram, FLM and RNNLM to the different Code-Switching attitudes.
In the case of n-gram and FLM, we build an n-gram LM or FLM with the clustered text for each attitude and interpolate this language model with the background n-gram or FLM using the interpolation weight 0.5. For the Code-Switching attitude dependent RNNLMs, we apply one-iteration re-training with a small learning rate as described in section 8.5.2. In all cases, we use the same parameter setup as for the Code-Switching attitude independent models. Table 8.9 summarizes the perplexities on the evaluation set speakers of our background LMs and also adapted LMs. We observe overall improvements by using the Code-Switching attitude dependent language models.

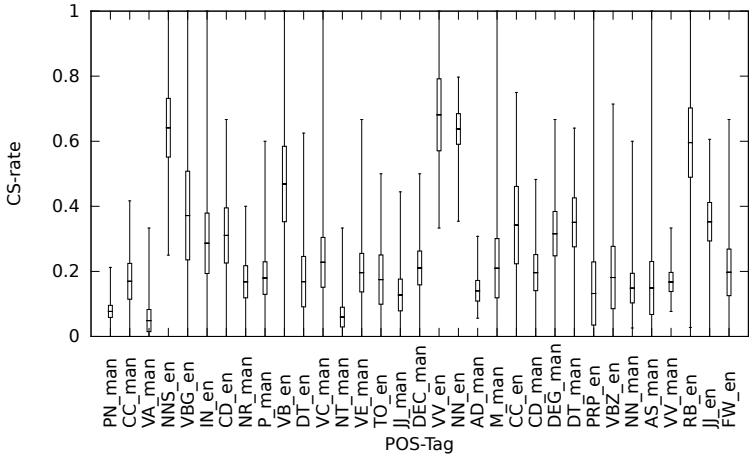


Figure 8.6: Distribution of speaker dependent Code-Switching rates after clustering in class 1

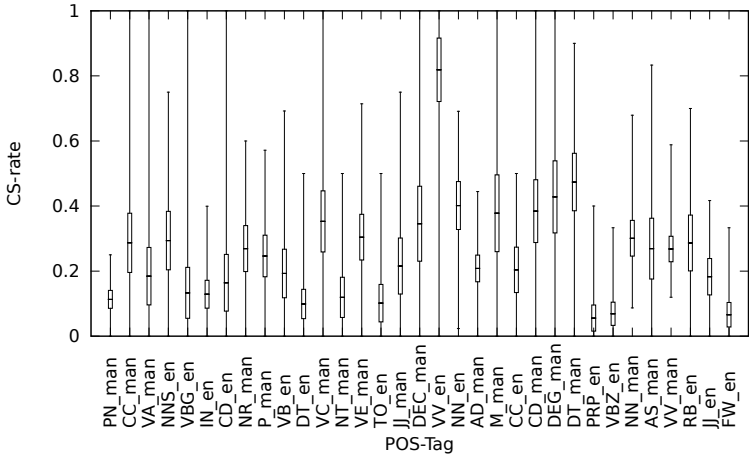


Figure 8.7: Distribution of speaker dependent Code-Switching rates after clustering in class 2

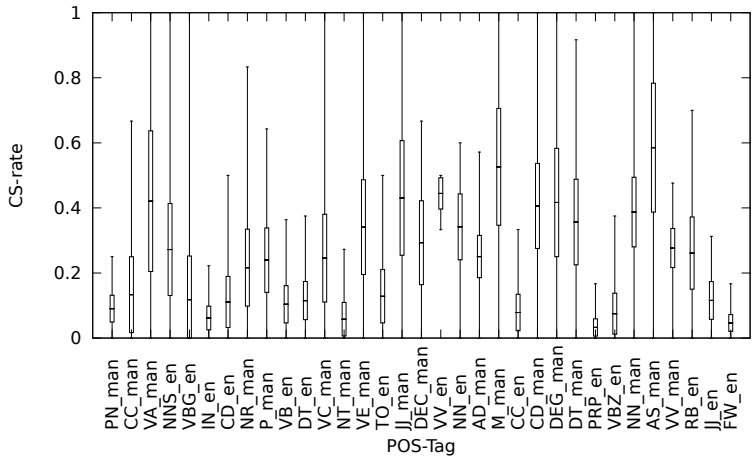


Figure 8.8: Distribution of speaker dependent Code-Switching rates after clustering in class 3

Table 8.9: *Perplexities of the Code-Switching dependent language models on the evaluation set speakers*

SPK	N-Gram	Adapted N-Gram	FLM	Adapted FLM	RNNLM	Adapted RNNLM
1	257.67	246.37	223.42	224.67	200.66	197.74
2	236.62	228.08	213.09	215.82	181.60	175.85
3	228.64	220.43	203.01	201.35	187.04	170.92
4	197.40	187.72	193.56	192.88	174.13	160.58
5	382.64	356.18	354.48	340.71	364.59	327.33
6	330.20	307.99	291.26	283.28	275.89	253.67
7	358.22	358.97	314.38	312.45	286.31	286.30
8	298.77	280.71	262.84	262.67	256.99	241.69

8.6 Rescoring Experiments

In this section, we present the experimental results achieved with our speech recognition system developed for the Code-Switching task.

8.6.1 Code-Switching ASR system

To decode the Code-Switching data of the SEAME corpus, we apply the speech recognition system (ASR) as described in [VLW+12]. This two-pass system first applies a speaker independent acoustic model which is trained with bottleneck

features. The second acoustic model is developed by applying Speaker Adaptive Training (SAT) with Feature Space Adaptation (FSA). To adapt to the Code-Switching problem and improve accuracy, language identity information is integrated into the decoding process using a multistream approach [WVT+12]. To obtain a dictionary, the CMU English [Dica] and Mandarin pronunciation dictionaries [HFT+08] are merged into one bilingual pronunciation dictionary. The number of English and Mandarin entries in the lexicon is 135k and 130k, respectively. Additionally, we apply several rules from [CTCL10] which might delete or change a phone to generate pronunciation variants for Singaporean English. On the language model side, the SRI Language Modeling Toolkit [Sto02] is used to build trigram language models from the SEAME training transcriptions containing all the words of the transcriptions. These models are interpolated with two monolingual language models that are created from 350k English sentences from NIST and 400k Mandarin sentences from the GALE project which have been collected from online newspapers. The vocabulary of 30k entries contains all the words of the transcriptions and the most frequent words of the monolingual corpora.
Furthermore, characteristics of Code-Switching from the SEAME training transcriptions are analyzed and additional Code-Switching text is generated artificially as described in [VLW+12]. The resulting language model has a perplexity of 483.9 and an out-of-vocabulary (OOV) rate of 1.21% on the SEAME development set transcriptions. This baseline system achieves an error rate of 35.5% MER on the SEAME development set.

8.6.2 ASR experiments using n-best rescoring

N-best rescoring We finally present the performance of each model in terms of mixed error rate when using it for rescoring. In these experiments, we rescore the 100-best lists of our ASR system with different settings for language model weights (lz) and word insertion penalties (lp). Equation 8.6 shows how the score for each hypothesis is computed. $|w|$ refers to the number of words in the hypothesis and λ to the interpolation weight of the recurrent neural network or factored language model lm_2. The decoding language model is denoted by lm_1. In our experiments, λ is set to 0.5.

$$\begin{aligned} score_{lm} &= \lambda \cdot score_{lm_2} + (1-\lambda) \cdot score_{lm_1} \\ score &= lz \cdot score_{lm} + score_{am} + lp \cdot |w| \end{aligned} \tag{8.6}$$

As performance measure, we have established the Mixed Error Rate (MER) which applies word error rates to English and character error rates to Mandarin segments [VLW+12]. Its result is the weighted average over all the English and Mandarin parts of the speech recognition output. By applying character based

error rates to Mandarin, the performance does not depend on the word segmentation algorithm for Mandarin. Thus, the performance can be compared across different segmentations. In this case, we use a manual word segmentation.
An interpolated language model *RNNLM (POS + LID) + FLM (POS + LID)* achieves the best rescoring result with a mixed error rate of 34.4% on the development set and an error rate of 29.2% on the evaluation set. This is an improvement of 3.1% and 2.7% relative to the baseline system as summarized in table 8.10.

Table 8.10: *MER(%) results of different models on the SEAME dev and test set*

Model	Dev set	Eval set
3-gram	35.5	30.0
RNNLM	35.6	29.3
RNNLM (LID)	34.9	29.4
RNNLM (POS)	34.8	29.3
RNNLM (POS + LID)	34.7	**29.2**
FLM (POS)	35.2	29.7
FLM (POS + LID)	35.2	29.7
RNNLM (POS + LID) + FLM (POS + LID)	**34.4**	**29.2**

Performance analysis We perform an analysis on the SEAME development set to investigate why the CS-LM performs better than the standard trigram model. The analysis shows that the trigram model recognizes 1889 Code-Switching points (41.11%) correctly, whereas the CS-LM detects 1990 language changes (43.31%) correctly. In addition, the CS-LM also outperforms the trigram model on monolingual segments. On English segments, it achieves a word error rate of 49.07%, while the trigram model has a word error rate of 50.21%. On Mandarin segments, the character error rates are 30.32% and 30.90%, respectively.

Adaptation results Finally, the adapted models are used for decoding and rescoring. In the rescoring process, we need to detect the class that fits best to each speaker (or utterance). We use the RNN language model score for this decision: We choose that class for rescoring that provides the best language model score. Hence, we show that the language model score for a specific speaker is correlated to the perplexities of the different clusters: The higher the score, the lower the perplexity. As correlation measure, the Pearson product-moment cor-

relation coefficient is used. It is shown in equation 8.7.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{8.7}$$

Table 8.11 reports the correlation values for each speaker and also in average. A correlation value of -1 means perfect negative correlation. A value of 0 means no correlation. The results indicate a very high correlation between the RNN language model score and the perplexity of the clustered classes. Therefore, we can use this score to indicate which adapted language model should be used for which speaker.

Table 8.11: *Correlation values between language model score per speaker and perplexity of the clustered classes (Spk abbreviates the work Speaker)*

Spk 1	Spk 2	Spk 3	Spk 4	Spk 5	Spk 6	Spk 7	Spk 8	average
-0.98	-1.0	-1.0	-1.0	-0.99	-1.0	-0.92	-0.96	-0.98

Table 8.12 shows the results on the SEAME development and evaluation set. Compared to the best Code-Switching independent LMs, we obtained an improvement of 0.4% absolute on both sets.

Table 8.12: *Mixed error rate results after decoding and rescoring with the adapted language models*

Model	Dev set	Eval set
SI N-Gram model	35.5%	30.0%
SI RNNLM + FLM	34.4%	29.2%
Adapted N-Gram + RNNLM + FLM	**34.0%**	**28.8%**

8.7 Summary

This chapter described our latest investigations of multilingual language modeling applied to Code-Switching speech. First, we presented an analysis of the manual transcriptions of the Code-Switching corpus SEAME to figure out which features might have a high potential to predict Code-Switching points. Then, we extended and applied recurrent neural network language models and factored language models to Code-Switching speech. Hence, we could integrate not only words but also both POS tags and language identification information into the models. The results showed that RNNLM and FLM outperform

the n-gram LM. Moreover, they provide complementary information as our backoff-level analysis showed. Hence, an interpolated language model integrating both language model types performed best on our database in terms of perplexity and also mixed error rate. Finally, we showed that Code-Switching is a speaker dependent phenomenon. Therefore, we clustered similar Code-Switching attitudes using cosine-distances and adapted our background language models using the corresponding training texts of these clusters. To sum up, Code-Switching attitude dependent language models provided reductions in terms of perplexity and also improvements in terms of mixed error rate.

CHAPTER 9

Conclusion and Future Directions

During the last decade it is noticeable that speech technology appears more frequently in the daily life in many different applications. Moreover, speech technology is required to be capable of handling multiple languages and multilingual challenges, such as non-native and Code-Switching speech. This thesis includes a wide range of research on multilingual speech recognition including multilingual feature extraction, multilingual acoustic modeling and multilingual language modeling. This chapter concludes the most important contributions and suggests potential future research directions.

9.1 Summary of the Thesis

The most important achievements of this thesis are structured in the following categories corresponding to the three most important applications of multilingual speech recognition.

- Rapid language adaptation to low-resource languages
- Adaptation to non-native speech
- ASR for Code-Switching speech

The contributions to the first aspect provide techniques to develop ASR systems for new languages with minimal time effort and reasonable costs. In contrast, the last two aspects are strongly related to multilingual challenges in which the linguistic characteristics of two languages impact each other. To sum up, the main achievement of this thesis is to investigate techniques which allow using resources from multiple resource-rich languages, such as data and models, to improve the ASR performance on low-resource languages and accents.

We hope that our research ideas presented in this thesis will have the potential to influence both practical applications and future research. The following sections summarize the most important results and show the importance of the thesis in the context of multilingual speech recognition.

9.1.1 ASR for low-resource languages using multilingual and crosslingual information

Building an ASR system without transcribed data One of the most important contributions of this thesis is the multilingual unsupervised training framework which allows building an ASR system without transcribed data. The idea is to use several existing ASR systems from different languages in combination with cross-language transfer techniques and unsupervised training to iteratively transcribe the audio data of the target language and, therefore, bootstrap an ASR system. The key contribution is the proposal of a word-based confidence score called "multilingual A-stabil". Compared to state-of-the-art confidence scores, "multilingual A-stabil" works well not only with well trained acoustic models but also with a poorly estimated acoustic model, such as one which is borrowed from other languages in order to bootstrap the acoustic model of an unseen language. To evaluate this framework, we conducted different experiments with increasing levels of difficulty. First, we developed a Czech ASR system without any transcribed training data using source languages which are related and non-related to the target language. Second, we applied our framework to Vietnamese using different European languages as source languages. All the experimental results showed that we are able to build an ASR system for a new language without any transcribed data, even if the source and the target languages are not related.

Multilingual Bottle-Neck features The integration of Multilayer Perceptron (MLP) features into ASR have become a state-of-the-art technique. Our exploration on multilingual bottle-neck features and their application to rapid language adaptation for a new language demonstrated their success in different tasks with different data sets. Our results revealed that using the multilingual

MLP to initialize the MLP training for new languages improved the MLP performance and, therefore, the ASR performance. The number of languages and the amount of data used to train the multilingual MLP has a strong effect on the MLP training and ASR performance. More source languages improve the MLP performance for a new language and also the ASR performance. Moreover, depending on the amount of training data of the target language, the language relation between source languages and target languages becomes important. If many training data are available, using related languages helps. In contrast, if only a small amount of training data is available, language similarity does not help but the number of source languages and the amount of data matters. Multilingual bottle-neck features are language independent and can be used for rapid language adaptation without re-training to improve the ASR performance. However, even with a very small amount of training data, the MLP re-training helps to improve the ASR performance. Visualization of the BN features using t-SNE shows potential to develop a better understanding of the multilingual BN features. Furthermore, the visualization suggests that the MLP seems to learn the *F1* and *F2* formants, which characterize different vowels, and to normalize their language dependent variations.

Multilingual deep neural network based acoustic modeling for rapid language adaptation This thesis comprises an investigation of multilingual deep neural network (DNN) based acoustic models and their application to new languages. We investigated the effect of phone merging on multilingual DNN in the context of rapid language adaptation. Moreover, the combination of multilingual DNNs with Kullback–Leibler divergence based acoustic modeling (KL-HMM) was explored. Using ten different languages from the GlobalPhone database, our studies revealed that crosslingual acoustic model transfer through multilingual DNNs was superior to unsupervised RBM pre-training and greedy layer-wise supervised training. We also found that KL-HMM based decoding consistently outperformed conventional hybrid decoding, especially in low-resource scenarios. Furthermore, the experiments indicated that multilingual DNNs training equally benefits from simple phone set concatenation and manually derived universal phone sets based on IPA.

9.1.2 Improving ASR for low-resource accents using multilingual and crosslingual information

Application of multilingual ASR to non-native speech This research presented our exploration of using multilingual and crosslingual information to improve the ASR performance on non-native speech. The study started with an investigation of the effect of multilingual acoustic modeling on non-native

speech. We showed that a bilingual L1-L2 acoustic model significantly improves the ASR performance. For the case that L1 is unknown or L1 data is not available, a multilingual ASR system consistently outperforms the monolingual L2 ASR system. The experimental results indicate that a multilingual acoustic model is more suitable to non-native speech than a monolingual acoustic model.

Improving ASR on non-native speech without adaptation data Finally, we proposed a method called *cross-lingual accent adaptation* to improve the ASR performance on non-native speech without any adaptation data. In our experiments, we applied English with Chinese accent to improve the German ASR on German with Chinese accent. Without using any adaptation data, we achieved a substantial improvement compared to the monolingual baseline system. This research work showed that it is possible to improve the ASR system on non-native speech without adaptation data, which has, to our knowledge, never been shown in the literature before.

9.1.3 Multilingual ASR for Code-Switching speech

In this thesis, we built an ASR system for conversational Mandarin-English Code-Switching speech. The system was trained with the SEAME database - a speech corpus which contained 65 hours of conversational speech recorded in Malaysia and Singapore. We focused on optimizing the language model for Code-Switching speech since it is one of the most important and also challenging components for this task.

Language model for Code-Switching speech Since the text data to build a robust language model is limited, we investigated the integration of high level features into the language model. First, a textual analysis was performed and the results illustrated that there are some trigger words and Part-Of-Speech tags after which people in Singapore and Malaysia tend to switch between Mandarin and English. Second, we integrated those features into state-of-the-art language modeling techniques, such as Recurrent Neural Network and Factored language models to improve the perplexity and mixed error rate on Code-Switching speech. Although RNNLMs generally outperformed the FLMs, our analysis revealed that FLMs were superior in the case of back-off to bigram. Hence, the interpolated language model between those two LMs gave the best performance on our database in terms of perplexity and also mixed error rate.

***Code-Switching attitude* dependent language modeling** Finally, we investigated the adaptation of Code-Switching language models to different speaker groups. Our textual analysis on speaker level revealed that Code-Switching is a speaker dependent phenomenon. Therefore, we clustered similar Code-Switching attitudes using cosine-distances. Afterwards, we adapted our N-Gram, FLM and RNN language model using the corresponding training texts of these clusters. We showed that this approach leads to further reductions in terms of perplexity and small improvements in terms of mixed error rate.

9.2 Potential Future Research Directions

This thesis addressed a wide range of research related to multilingual speech recognition. Its results have a potential to make an impact on future practical applications and research. However, to improve speech technology for daily life applications and to increase their usage all over the world, further research needs to be conducted in the future. This section suggests three different research directions related to low-resource languages, and multilingual challenges, such as non-native and Code-Switching speech.

9.2.1 Unwritten languages

All over the world, there are around 6,900 languages including resource-rich and low-resource languages. This thesis addressed the challenges of building ASR systems for low-resource languages which lack resources for speech and language technology. However, if all the languages in the world should be investigated, methods to deal with languages without writing systems are required in the future. Currently in this context, there is a number of interesting ongoing studies ([BZG06, SBW09, SSVS12, SSVS13]) on the automatic discovery of vocabulary and the corresponding pronunciation dictionary for unwritten languages. Moreover, speech synthesis for unwritten languages [SPC+13] gains a lot of attention of the speech community. In the future, it is interesting to combine all the techniques to develop a speech-to-speech translation system for unwritten languages.

9.2.2 ASR for native and non-native speech

Non-native speech recognition has become one of the most important and challenging applications of ASR due to the rapid growth of globalization. Most of the research on non-native speech including our investigation were performed

in order to improve the ASR performance on non-native speech without considering the effect on native speech. Multilingual and adaptation data were used to train a strong multilingual acoustic model and adapt it to the target accent. However, those techniques lower the performance on native speech. In practical applications, systems using speech technology aim at running in speaker independent mode. Hence, it is not clear whether the speaker is a native or a non-native speaker. Therefore, techniques are necessary which improve the ASR performance on native and non-native speech at the same time.

9.2.3 Research on Code-Switching speech

Bilingualism has become more common in many different countries, such as Singapore, Malaysia, South Africa, USA or India. Therefore, ASR for Code-Switching speech will gain more attention in the speech research community. In this thesis, we conducted research on ASR for Code-Switching, especially on language modeling. The experimental results are limited since the newly developed methods were only evaluated on Mandarin-English Code-Switching speech. Obviously, one of the most important lack is missing Code-Switching speech databases with different language combinations. It complicates research work and the drawing of a general conclusion for all Code-Switching situations. In the future, promising research methods on Code-Switching speech should be investigated and evaluated across Code-Switching databases. It would be interesting to explore not only the effect by changing the language combination but also the difference among different geographical areas where the same language combinations are used.

Although the list of potential future works is surely not complete, we hope that it provides researchers with advice for further investigations of multilingual speech recognition.

Bibliography

[AHG+98] M. Aretoulaki, S. Harbeck, F. Gallwitz, E. Nöth, H. Niemann, J. Ivaneckỳ, I. Ipsic, N. Pavesic, and V. Matousek. SQEL: a multilingual and multifunctional dialogue system. In *Proc. of ICSLP*, 1998.

[Ass99] International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[Aue99a] P. Auer. *Code-switching in conversation*. Routledge, 1999.

[Aue99b] P. Auer. From codeswitching via language mixing to fused lects toward a dynamic typology of bilingual speech. *International Journal of Bilingualism*, 3(4):309–332, 1999.

[AVK+12] H. Adel, N.T. Vu, F. Kraus, T. Schlippe, T. Schultz, and H. Li. Recurrent neural network language modeling for Code Switching conversational speech. In *Proc. of ICASSP*, pages 8411–8415, 2012.

[BBH+00] W. Byrne, P. Beyerlein, J.M. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Peterek, J. Picone, D. Vergyri, and T. Wang. Towards language independent acoustic modeling. In *Proc. of ICASSP*, pages 1029–1032, 2000.

[BFI+06] G. Bouselmi, D. Fohr, I. Illina, J.-P. Haton, et al. Multilingual non-native speech recognition using phonetic confusion-based acoustic model modification and graphemic constraints. In *Proc. of ICSLP*, pages 599–603, 2006.

[BK03] J.A. Bilmes and K. Kirchhoff. Factored language models and generalized parallel backoff. In *Proc. of HLT-NAACL*, pages 4–6, 2003.

[Bla11] F. Blaicher. Smt-based text generation for Code-Switching language models. Master's thesis, Cognitive Systems Lab (CSL), Karlsruhe Insitutite of Technology (KIT), 2011.

[BLPL07] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[BM94] H. Bourlard and N. Morgan. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer, 1994.

[Bok89] E.G. Bokamba. Are there syntactic constraints on code-mixing? *World Englishes*, 8(3):277–292, 1989.

[BSA+10] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, M. Goel, N.vand Karafiát, D. Povey, et al. Multilingual acoustic modeling for speech recognition based on subspace Gaussian Mixture Models. In *Proc. of ICASSP*, pages 4334–4337, 2010.

[Bur10] C. Burgmer. Detecting code-switch events based on textual features. Diploma thesis, Cognitive Systems Lab (CSL), Karlsruhe Insitutite of Technology (KIT), 2010.

[BZG06] L. Besacier, B. Zhou, and Y. Gao. Towards speech translation of non written languages. In *Proc. of SLT*, pages 222–225, 2006.

[CC97] A. Constantinescu and G. Chollet. On cross-language experiments and data-driven units for alisp (automatic language independent speech processing). In *Proc. of ASRU*, pages 606–613, 1997.

[CCLC06] J.Y.C. Chan, PC Ching, T. Lee, and H. Cao. Automatic speech recognition of Cantonese-English code-mixing utterances. In *Proc. of ICSLP*, 2006.

[CDG+97] P. Cohen, S. Dharanipragada, J. Gros, M. Monkowski, C. Neti, S. Roukos, and T. Ward. Towards a universal speech recognizer for multiple languages. In *Proc. of ASRU*, pages 591–598. IEEE, 1997.

[CMDL+07] O. Cetin, M. Magimai-Doss, K. Livescu, A. Kantor, S. King, C. Bartels, and J. Frankel. Monolingual and crosslingual comparison of tandem features derived from articulatory and phone MLPs. In *Proc. of ASRU*, pages 36–41, 2007.

[CTCL10] W. Chen, Y. Tan, E.S. Chng, and H. Li. The development of a Singapore English call resource. *Proc. of Oriental COCOSDA*, 2010.

[DC97] M. Díaz-Campos. The effects of formal instruction on the acquisition of Spanish stop consonants. *Contemporary Perspectives on the Acquisition of Spanish*, 2(2):57 – 75, 1997.

[Dev85] P.A. Devijver. Baum's forward-backward algorithm revisited. *Pattern Recognition Letters*, 3(6):369–373, 1985.

[Dica] CMU Dictionary. www.speech.cs.cmu.edu/cgi-bin/cmudict.

[Dicb] Leipzig Vietnamese Pronunciation Dictionary. http://www.informatik.uni-leipzig-de/ duc/dict/install.htm.

[Dic05] The New Oxford American Dictionary. Second edition. 2005.

[DK04] K. Duh and K. Kirchhoff. Automatic learning of language model structure. In *Proc. of the 20th International Conference on Computational Linguistics*, page 148, 2004.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[DM80] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

[DVBD+11] N. De Vries, J. Badenhorst, M. Davel, E. Barnard, and A. De Waal. Woefzela-an open-source platform for asr data collection in the developing world. In *Proc. of Interspeech*, pages 3177–3180, 2011.

[DYDA12] G.E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

[EBC+10] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, pages 625–660, 2010.

[EDSN10] A. El-Desoky, R. Schlüter, and H. Ney. A hybrid morphologically decomposed factored language models for Arabic LVCSR. In *Proc. of HLT*, pages 701–704, 2010.

[ELR12] ELRA. European language resources association (ELRA), http://catalog.elra.info, Retrieved November 30, 2012.

[EW00] G. Evermann and P. Woodland. Large vocabulary decoding and confidence estimation using word posterior probabilities. In *Proc. of ICASSP*, volume 3, pages 1655–1658, 2000.

[FFN97] J.E. Flege, E.M. Frieda, and T. Nozawa. Amount of native-language (l1) use affects the pronunciation of an l2. *Journal of Phonetics*, 25(2):169–186, 1997.

[FGH+97] M. Finke, P. Geutner, H. Hild, T. Kemp, T. Ries, and M. Westphal. The Karlsruhe-Verbmobil speech recognition engine. In *Proc. of ICASSP*, pages 83–86, 1997.

[FJ73] G.D. Forney Jr. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[Fle80] J.E. Flege. Phonetic approximation in second language acquisition. *Language Learning*, 30(1):117–134, 1980.

[Fle87] J.E. Flege. The production of new and similar phones in a foreign language: Evidence for the effect of equivalence classification. *Journal of phonetics*, 15(1):47–65, 1987.

[FSS+03] C. Fügen, S. Stüker, H. Soltau, F. Metze, and T. Schultz. Efficient handling of multilingual language models. In *Proc. of ASRU*, pages 441–446, 2003.

[Fuk90] K. Fukunaga. *Introduction to statistical pattern recognition*. Access Online via Elsevier, 1990.

[Gal98] M. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2):75–98, 1998.

[Gal99] M. Gales. Semi-tied covariance matrices for hidden markov models. *IEEE Transactions on Speech and Audio Processing*, 7(3):272–281, 1999.

[GE03] S. Goronzy and K. Eisele. Automatic pronunciation modelling for multiple non-native accents. In *Proc. of ASRU*, pages 123–128, 2003.

[GG97] S. Gokcen and J.M. Gokcen. A multilingual phoneme and model set: toward a universal base for automatic speech recognition. In *Proc. of ASRU*, 1997.

[GI90] T.V. Gamkrelidze and V.V. Ivanov. The early history of Indo-European languages. *Scientific American*, pages 110–116, 1990.

[GKKC07] F. Grézl, M. Karafiát, S. Kontár, and J.H. Cernocky. Probabilistic and bottle-neck features for lvcsr of meetings. In *Proc. of ICASSP*, pages IV–757, 2007.

[GL94] J.L. Gauvain and C.H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of markov chains. *IEEE transactions on Speech and audio processing*, 2(2):291–298, 1994.

[Gor] R. G. Gordon. *Ethnologue: Languages of the World*, volume 15. Dallas: SIL International.

[GSR13] A. Ghoshal, P. Swietojanski, and S. Renals. Multilingual training of deep neural networks. In *Proc. of ICASSP*, pages 7319–7323, 2013.

[HAH01] X. Huang, A. Acero, and H.W. Hon. *Spoken language processing*, volume 15. Prentice Hall PTR New Jersey, 2001.

[HCC04] C. Huang, T. Chen, and E. Chang. Accent issues in large vocabulary continuous speech recognition. *International Journal of Speech Technology*, 7(2-3):141–153, 2004.

[HDS00] H. Hermansky, P.W.E. Daniel, and S. Sangita. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. of ICASSP*, pages 1635–1638, 2000.

[Her90] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87:1738, 1990.

[HFT+08] R. Hsiao, M. Fuhs, Y.C. Tam, Q. Jin, and T. Schultz. The CMU-interACT 2008 Mandarin transcription system. In *Proc. of Interspeech*, pages 1445–1448, 2008.

[HHA96] M.-Y. Hwang, X. Huang, and F. Alleva. Predicting unseen triphones with senones. *IEEE Transctions on Speech and Audio Processing*, 4(6):412–419, 1996.

[HLY+13] J.T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Proc. of ICASSP*, pages 7304–7308, 2013.

[HOT06] G.E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[HR02] G.E. Hinton and Sam T. Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, pages 833–840, 2002.

[htt12] Appen Butler Hill Pty Ltd http://www.appen.com. Speech and language resources, 2012.

[HVS+13] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean. Multilingual acoustic models using distributed deep neural networks. In *Proc. of ICASSP*, pages 8619–8623, 2013.

[HWP96] J.J. Humphries, P.C. Woodland, and D. Pearce. Using accent-specific pronunciation modelling for robust speech recognition. In *Proc. of ICSLP*, pages 2324–2327, 1996.

[IBD10] D. Imseng, H. Bourlard, and M.M. Doss. Towards mixed language speech recognition systems. In *Proc. of Interspeech*, pages 278–281, 2010.

[IKHZ00] B. Imperl, Z. Kacic, B. Horvat, and A. Zgank. Agglomerative vs. tree-based clustering for the definition of multilingual set of triphones. In *Proc. of ICASSP*, pages 1273–1276, 2000.

[IMBG13] D. Imseng, P. Motlicek, H. Bourlard, and P. N. Garner. Using out-of-language data to improve an under-resourced speech recognizer. *Speech Communication*, pages 142 – 151, 2013.

[IMGB13] D. Imseng, P. Motlicek, P. N. Garner, and H. Bourlard. Impact of deep MLP architecture on different acoustic modeling techniques for under-resourced speech recognition. In *Proc. of ASRU*, 2013.

[Jel05] F. Jelinek. Some of my best friends are linguists. *Language resources and evaluation*, 39(1):25–34, 2005.

[Jia05] H. Jiang. Confidence measures for speech recognition: A survey. *Speech Communication*, 45(4):455–470, 2005.

[JVS12] M.J. Jose, N.T. Vu, and T. Schultz. Initial experiments with Tamil LVCSR. In *Proc. of Asian Language Processing (IALP)*, pages 81–84, 2012.

[JVS13] M.J. Jose, N.T. Vu, and T. Schultz. Experiments towards a better LVCSR system for Tamil. In *Proc. of Interspeech*, 2013.

[Kat02] K. Katzner. *The languages of the world*. Routledge, 2002.

[KK02] W. Kim and S. Khudanpur. Using cross-language cues for story-specific language modeling. In *Proc. of Interspeech*, 2002.

[KMKB11] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget. Recurrent neural network based language modeling in meeting recognition. *Proc. of Interspeech*, pages 2877–2880, 2011.

[Koh96] J. Kohler. Multi-lingual phoneme recognition exploiting acoustic-phonetic similarities of sounds. In *Proc. of ICSLP*, pages 2195–2198, 1996.

[Köh98] J. Köhler. Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks. In *Proc. of ICASSP*, pages 417–420, 1998.

[Kra11] F. Kraus. Cross-language bootstrapping based on completely unsupervised training. Master's thesis, Cognitive Systems Lab (CSL), Karlsruhe Insitutite of Technology (KIT), 2011.

[KS97] T. Kemp and T. Schaaf. Estimating confidence using word lattices. In *Proc. of Eurospeech*, pages 827–830, 1997.

[KSW99] D. Kiecza, T. Schultz, and A. Waibel. Data-driven determination of appropriate dictionary units for Korean LVCSR. In *Proc. of ICASSP*, pages 323–327, 1999.

[Kul87] S. Kullback. The kullback-leibler distance. *The American Statistician*, 41(4):340–341, 1987.

[KW99] T. Kemp and A. Waibel. Unsupervised training of a speech recognizer: recent experiments. In *Proc. of Eurospeech*, 1999.

[LB12] LM-BM. Benchmark globalphone language models, http://csl.ira.uka.de/globalphone, Retrieved November 30, 2012.

[LBS06] V.B. Le, L. Besacier, and T. Schultz. Acoustic-phonetic unit similarities for context dependent acoustic model portability. In *Proc. of ICASSP*, page I, 2006.

[LDY+09] H. Lin, L. Deng, D. Yu, Y. Gong, A. Acero, and C.H. Lee. A study on multilingual acoustic modeling for large vocabulary ASR. In *Proc. of ICASSP*, pages 4333–4336, 2009.

[Lee88] K.-F. Lee. On large-vocabulary speaker-independent continuous speech recognition. *Speech Communication*, 7(4):375–379, 1988.

[LG97] R. Lippi-Green. *English with an Accent: Language, Ideology, and Discrimination in the United States*. New York: Routledge, 1997.

[LGA02a] L. Lamel, J.L. Gauvain, and G. Adda. Lightly supervised and unsupervised acoustic model training. *Computer Speech & Language*, 16(1):115–129, 2002.

[LGA02b] L. Lamel, J.L. Gauvain, and G. Adda. Unsupervised acoustic model training. In *Proc. of ICASSP*, pages I–877, 2002.

[LGE+91] L. Lamel, J.L. Gauvain, M. Eskenazi, et al. BREF, a large vocabulary spoken corpus for french. 1991.

[LGN09] J. Lööf, C. Gollan, and H. Ney. Cross-language bootstrapping for unsupervised acoustic model training: Rapid development of a polish speech recognition system. In *Proc. of Interspeech*, pages 88–91, 2009.

[LHH+90] K.F. Lee, S. Hayamizu, H.W. Hon, C. Huang, J. Swartz, and R. Weide. Allophone clustering for continuous speech recognition. In *Proc. of ICASSP*, pages 749–752, 1990.

[Liv99] K. Livescu. *Analysis and modeling of non-native speech for automatic speech recognition*. PhD thesis, Massachusetts Institute of Technology (MIT), 1999.

[LM06] A. Lakshmi and H.A. Murthy. A syllable based continuous speech recognizer for Tamil. In *Proc. of Interspeech*, 2006.

[LRN88] J. Lee Rodgers and W.A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[LTCL10] D.C. Lyu, T.P. Tan, E.S. Chng, and H. Li. An analysis of a Mandarin-English Code-switching speech corpus: SEAME. *Proc. of Interspeech*, pages 25–28, 2010.

[LW95] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech & Language*, 9(2):171–185, 1995.

[MDH12] A. Mohamed, G.E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22, 2012.

[MHJ+10] F. Metze, R. Hsiao, Q. Jin, U. Nallasamy, and T. Schultz. The 2010 CMU GALE speech-to-text system. In *Proc. of Interspeech*, pages 1501–1504, 2010.

[Mih11] Z. Mihaylova. Lexical and acoustic adaptation for multiple non-native english accents. Master's thesis, Cognitive Systems Lab (CSL), Karlsruhe Insitutite of Technology (KIT), 2011.

[MKB+10] T. Mikolov, M. Karafiát, L. Burget, J.H. Cernocky, and S. Khudanpur. Recurrent neural network based language model. In *Proc. of Interspeech*, pages 1045–1048, 2010.

[MKB+11] T. Mikolov, S. Kombrink, L. Burget, J.H. Cernocky, and S. Khudanpur. Extensions of recurrent neural network language model. In *Proc. of ICASSP*, pages 5528–5531, 2011.

[MKD+11] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernockỳ. RNNLM–recurrent neural network language modeling toolkit. In *Proc. of ASRU*, pages 196–201, 2011.

[MMS93] M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19:313–330, 1993.

[Muy00] P. Muysken. *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press, 2000.

[Nol64] A.M. Noll. Short-time spectrum and cepstrum techniques for vocal-pitch detection. *The Journal of the Acoustical Society of America*, 36:296, 1964.

[NRRW11] F. Niu, B. Recht, C. Ré, and S.J. Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *arXiv preprint arXiv:1106.5730*, 2011.

[OSNG12] I. Oparin, M. Sundermeyer, H. Ney, and J.L. Gauvain. Performance analysis of neural networks in combination with n-gram language models. In *Proc. of ICASSP*, pages 5005–5008, 2012.

[PB92] D.B. Paul and J.M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Proc. of the Workshop on Speech and Natural Language*, pages 357–362, 1992.

[PBA+10] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N.K. Goel, M. Karafiát, A. Rastrow, et al. Subspace gaussian mixture models for speech recognition. In *Proc. of ICASSP*, pages 4330–4333, 2010.

[PGB+11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, and P. Schwarz. The Kaldi speech recognition toolkit. In *Proc. of ASRU*, 2011.

[PKK+08] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. Boosted mmi for model and feature-space discriminative training. In *Proc. of ICASSP*, pages 4057–4060, 2008.

[Pop78] S. Poplack. *Syntactic structure and social function of code-switching*, volume 2. Centro de Estudios Puertorriqueños, University of New York, 1978.

[Pop80] S. Poplack. Sometimes i'll start a sentence in spanish y termino en español: toward a typology of code-switching. *Linguistics*, 18(7-8):581–618, 1980.

[PSN11] C. Plahl, R. Schlüter, and H. Ney. Cross-lingual portability of Chinese and English neural network features for French and German LVCSR. In *Proc. of ASRU*, pages 371–376, 2011.

[QN] QN. http://www.icsi.berkeley.edu/speech/qn.html.

[QPL11] Y. Qian, D. Povey, and J. Liu. State-level data borrowing for low-resource speech recognition based on subspace GMMs. In *Proc. of Interspeech*, pages 553–560, 2011.

[Rab89] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[RGN08] M. Raab, R. Gruhn, and E. Nöth. Multilingual weighted codebooks for non-native speech recognition. In *Text, Speech and Dialogue*, pages 485–492, 2008.

[RHW02a] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1:213, 2002.

[RHW02b] D. Rumelhart, G. E Hinton, and R. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1:213, 2002.

[RLA12] RLAT. Rapid language adaptation toolkit (RLAT), http://csl.ira.uka.de/rlat-dev, Retrieved November 30 2012.

[Ros00] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

[SBB+07] T. Schultz, A. Black, S. Badaskar, M. Hornyak, and J. Kominek. Spice: Web-based tools for rapid language adaptation in speech processing systems. In *Interspeech*, 2007.

[SBW09] S. Stüker, L. Besacier, and A. Waibel. Human translations guided language discovery for asr systems. In *Proc. of Interspeech*, pages 3023–3026, 2009.

[Sch98] Kjell Schubert. Pitch tracking and his application on speech recognition. Master's thesis, University of Karlsruhe, 1998.

[Sch02] T. Schultz. Globalphone: a multilingual speech and text database developed at Karlsruhe University. In *Proc. of ICSLP*, pages 345–348, 2002.

[SDV+12] T. Schlippe, E.G.K. Djomgang, N.T. Vu, S. Ochs, and T. Schultz. Hausa large vocabulary continuous speech recognition. In *Proc. of SLTU*, 2012.

[SGR12] P. Swietojanski, A. Ghoshal, and S. Renals. Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR. In *Proc. of SLT*, pages 246–251, 2012.

[SK97] T. Schaaf and T. Kemp. Confidence measures for spontaneous speech recognition. In *Proc. of ICASSP*, volume 2, pages 875–878, 1997.

[SK06] T. Schultz and K. Kirchhoff. *Multilingual speech processing*. Access Online via Elsevier, 2006.

[SKK+00] M. Steinbach, G. Karypis, V. Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526, 2000.

[SKW97a] A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error minimization in n-best list rescoring. In *Proc. of Eurospeech*, pages 163–166, 1997.

[SKW97b] A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error minimization in n-best list rescoring. In *Proc. of Eurospeech*, pages 163–166, 1997.

[SL08a] T. Solorio and Y. Liu. Learning to predict code-switching points. In *Proc. of the Conference on Empirical Methods in NLP*, pages 973–981, 2008.

[SL08b] T. Solorio and Y. Liu. Part-of-speech tagging for English-Spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in NLP*, pages 1051–1060, 2008.

[SLCY11] F. Seide, G. Li, X. Chen, and D. Yu. Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *Proc. of ASRU*, pages 24–29, 2011.

[SLY11] F. Seide, G. Li, and D. Yu. Conversational speech transcription using context-dependent deep neural networks. In *Proc. of Interspeech*, pages 437–440, 2011.

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

[SPC+13] S. Sitaram, S. Palkar, Y.-N. Chen, A. Parlikar, and A. Black. Bootstrapping text-to-speech for speech processing in languages without an orthography. In *Proc. of ICASSP*, pages 7992–7996, 2013.

[SSVS12] F. Stahlberg, T. Schlippe, S. Vogel, and T. Schultz. Word segmentation through cross-lingual word-to-phoneme alignment. In *Proc. of SLT*, pages 85–90, 2012.

[SSVS13] F. Stahlberg, T. Schlippe, S. Vogel, and T. Schultz. Pronunciation extraction from phoneme sequences through cross-lingual word-to-phoneme alignment. In *Proc. of SLSP*, 2013.

[Sto02] A. Stolcke. SRILM-an extensible language modeling toolkit. In *Proc. of SLP*, pages 901–904, 2002.

[SVS13] T. Schultz, N.T. Vu, and T. Schlippe. GlobalPhone: A multilingual text & speech database in 20 languages. In *Proc. of ICASSP*, pages 8126–8130, 2013.

[SW98a] T. Schultz and A. Waibel. Adaptation of pronunciation dictionaries for recognition of unseen languages. In *Proc. of SPIIRAS International Workshop on Speech and Computer*, pages 207–210, 1998.

[SW98b] T. Schultz and A. Waibel. Language independent and language adaptive large vocabulary speech recognition. In *Proc. of ICSLP*, pages 1819–1822, 1998.

[SW01a] T. Schultz and A. Waibel. Experiments on cross-language acoustic modeling. In *Proc. of Eurospeech*, pages 2721–2724, 2001.

[SW01b] T. Schultz and A. Waibel. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35(1):31–51, 2001.

[t-S] t-SNE. http://homepage.tudelft.nl/19j49/t-sne.html.

[TB07] T.-P. Tan and L. Besacier. Acoustic model interpolation for non-native speech recognition. In *Proc. of ICASSP*, pages IV–1009, 2007.

[TFGK08] L. Tóth, J. Frankel, G. Gosztolya, and S. King. Cross-lingual portability of MLP-based tandem features–a case study for English and Hungarian. In *Proc. of Interspeech*, pages 2695–2698, 2008.

[TGH06] S. Thomas, S. Ganapathy, and H. Hermansky. Cross-domain and cross-lingual portability of acoustic features estimated by multilayer perceptrons. In *Proc. of ICASSP*, page I, 2006.

[TGH10] S. Thomas, S. Ganapathy, and H. Hermansky. Cross-lingual and multi-stream posterior features for low resource LVCSR systems. In *Proc. of Interspeech*, pages 877–880, 2010.

[TGH12] S. Thomas, S. Ganapathy, and H. Hermansky. Multilingual MLP features for low-resource LVCSR systems. In *Proc. of ICASSP*, pages 4269–4272, 2012.

[TKMS03] K. Toutanova, D. Klein, C.D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT*, pages 173–180, 2003.

[TM00] K. Toutanova and C.D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proc. of the Joint SIGDAT conference on Empirical methods in NLP and very large corpora*, pages 63–70, 2000.

[Tom00a] L.M. Tomokiyo. Handling non-native speech in LVCSR: A preliminary study. In *Proc. of the EUROCALL/CALICO/ISCA workshop on Integrating Speech Technology in (Language) Learning*, 2000.

[Tom00b] L.M. Tomokiyo. *Lexical and acoustic modeling of non-native speech in LVCSR*. PhD thesis, Carnegie Mellon University (CMU), 2000.

[TW03] L.M. Tomokiyo and A. Waibel. Adaptation methods for non-native speech. *Multilingual Speech and Language Processing*, page 6, 2003.

[UNE13] UNESCO. http://www.unesco.org/new/en/culture/themes/endangered-languages/, Retrieved November 14th, 2013.

[UNGH98] N. Ueda, R. Nakano, Z. Ghahramani, and G.E. Hinton. Split and merge EM algorithm for improving Gaussian mixture density estimates. In *Proc. of the IEEE Signal Processing Society Workshop*, pages 274–283, 1998.

[VdMH08] L. Van der Maaten and G.E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[VGBP13] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey. Sequence-discriminative training of deep neural networks. In *Proc. of Interspeech*, 2013.

[Vit67] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[VKG+12] K. Vesely, M. Karafiát, F. Grezl, M. Janda, and E. Egorova. The language-independent bottleneck features. In *Proc. of SLT*, pages 336–341, 2012.

[VLBM08] P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[VLW+12] N.T. Vu, D.C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.S. Chng, T. Schultz, and H. Li. A first speech recognition system for Mandarin-English Code-Switch conversational speech. In *Proc. of ICASSP*, pages 4889–4892, 2012.

[VNLH05] T.T. Vu, D.T. Nguyen, M.C. Luong, and J.P. Hosom. Vietnamese large vocabulary continuous speech recognition. In *Proc. of Eurospeech*, 2005.

[VS09] N.T. Vu and T. Schultz. Vietnamese large vocabulary continuous speech recognition. In *Proc. of ASRU*, pages 333–338, 2009.

[VSKS10] N.T. Vu, T. Schlippe, F. Kraus, and T. Schultz. Rapid bootstrapping of five eastern european languages using the rapid language adaptation toolkit. In *Proc. of Interspeech*, pages 865–868, 2010.

[Wan13] Y. Wang. Crosslingual accent adaptation: A study case for English and German with Chinese accent. Bachelor's thesis, Cognitive Systems Lab (CSL), Karlsruhe Insitutite of Technology (KIT), 2013.

[WBNS97] F. Weng, H. Bratt, L. Neumeyer, and A. Stolcke. A study of multilingual speech recognition. In *Proc. of Eurospeech*, pages 359–362, 1997.

[Wer90] P.J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[Wik13] Wikipedia. http://en.wikipedia.org/wiki/File:-Primary_Human_Language_Families_Map.png, Retrieved on December 2nd 2013.

[WKAM94] B. Wheatley, K. Kondo, W. Anderson, and Y. Muthusamy. An evaluation of cross-language adaptation for rapid HMM development in a new language. In *Proc. of ICASSP*, pages I–237, 1994.

[WMN99] F. Wessel, K. Macherey, and H. Ney. A comparison of word graph and N-best list based confidence measures. In *Proc. of Eurospeech*, pages 315–318, 1999.

[WN05] F. Wessel and H. Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(1):23–31, 2005.

[WRN+98] T. Ward, S. Roukos, C. Neti, J. Gros, M. Epstein, and S. Dharanipragada. Towards speech understanding across multiple languages. In *Proc. of ICSLP*, 1998.

[WSW03] Z. Wang, T. Schultz, and A. Waibel. Comparison of acoustic model adaptation techniques on non-native speech. In *Proc. of ICASSP*, pages I–540, 2003.

[WVT+12] J. Weiner, N.T. Vu, D. Telaar, F. Metze, T. Schultz, D.C. Lyu, E.S. Chng, and H. Li. Integration of language identification into a recognition system for spoken conversations containing code-switches. In *Proc. of SLTU*, 2012.

[XXCP05] N. Xue, F. Xia, F.D. Chiou, and M. Palmer. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207, 2005.

[YPC12] S. Yangyang, W. Pascal, and M. J. Catholijn. Towards recurrent neural networks language models with linguistic and contextual features. *Proc. of Interspeech*, 2012.

[ZC98] G. Zavaliagkos and T. Colthurst. Utilizing untranscribed training data to improve performance. In *DARPA Broadcast News Transcription and Understanding Workshop*, pages 301–305, 1998.

[ZTPK14] X. Zhang, J. Trmal, P. Povey, and S. Khudanpur. Improving deep neural network acoustic models using generalized maxout networks. In *Proc. of ICASSP*, 2014.

Publications

Journal Articles

- Ngoc Thang Vu, Heike Adel, and Tanja Schultz. Language Modeling for Code-Switching Speech. *Special Issue: SLSP 2013*, *submitted for review*.

Peer-Reviewed Conference Publications

2014

- Ngoc Thang Vu, Jochen Weiner, Tanja Schultz. Investigating the Learning Effect of Multilingual Bottle-Neck Features for ASR. Interspeech 2014, *submitted for review*.
- Heike Adel, Katrin Kirchhoff, Ngoc Thang Vu, Dominic Telaar and Tanja Schultz. Comparing Approaches to Convert Recurrent Neural Networks into Backoff Language Models For Efficient Decoding. Interspeech 2014, *submitted for review*.
- Heike Adel, Dominic Telaar, Ngoc Thang Vu, Katrin Kirchhoff and Tanja Schultz. Combining Recurrent Neural Networks and Factored Language Models During Decoding of Code-Switching Speech. Interspeech 2014, *submitted for review*.
- Ngoc Thang Vu, Yuanfan Wang, Marten Klose, Zlatka Mihaylova, and Tanja Schultz. Improving ASR for Non-Native Speech using Multilingual and Crosslingual Information. Interspeech 2014, *submitted for review*.
- Dominic Telaar, Michael Wand, Dirk Gehrig, Felix Putze, Christoph Amma, Dominic Heger, Thang Vu, Mark Erhardt, Tim Schlippe, Matthias Janke, Christian Herff, and Tanja Schultz. BIOKIT - Real-Time Decoder for Biosignal Processing Interspeech 2014, *submitted for review*.
- Heike Adel, Katrin Kirchhoff, Dominic Telaar, Ngoc Thang Vu, Tim Schlippe, and Tanja Schultz. Features for Factored Language Models for Code-Switching speech. In *Proc. SLTU*, 2014.

- Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Molicek, Tanja Schultz, and Herve Boulard. Multilingual Deep Neural Network For Rapid Language Adaptation. In *Proc. ICASSP*, 2014.

2013

- Ngoc Thang Vu, and Tanja Schultz. Multilingual Multilayer Perceptron For Rapid Language Adaptation Between and Across Language Families. In *Proc. Interspeech*, 2013.
- Melvin Jose Johnson Premkumar, Ngoc Thang Vu, and Tanja Schultz. Experiments towards a better LVCSR System for Tamil. In *Proc. Interspeech*, 2013.
- Tim Schlippe, Lukasz Gren, Ngoc Thang Vu, and Tanja Schultz. Unsupervised Language Model Adaptation For Automatic Speech Recognition of Broadcast News Using WEB 2.0. In *Proc. Interspeech*, 2013.
- Heike Adel, Ngoc Thang Vu, and Tanja Schultz. Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling. In *Proc. ACL*, 2013.
- Ngoc Thang Vu, Heike Adel and Tanja Schultz. An Investigation of Code-Switching Attitude Dependent Language Modeling. In *Proc. SLSP*, 2013.
- Heike Adel, Ngoc Thang Vu, and Tanja Schultz. Recurrent Neural Network Language Modeling for Code Switching Conversational Speech. In *Proc. ICASSP*, 2013.
- Tanja Schultz, Ngoc Thang Vu, and Tim Schlippe. GlobalPhone: A Multilingual Text and Speech Database in 20 Languages. In *Proc. ICASSP*, 2013.

2012

- Melvin Jose J, Ngoc Thang Vu, and Tanja Schultz. Initial Experiments with Tamil LVCSR. In *Proc. IALP*, 2012.
- Ngoc Thang Vu, Wojtek Breiter, Florian Metze and Tanja Schultz. An Investigation on Initialization Schemes for Multilayer Perceptron Training Using Multilingual Data and their Effect on ASR Performance. In *Proc. Interspeech*, 2012.
- Tim Schlippe, Sebastian Ochs, Ngoc Thang Vu and Tanja Schultz. Automatic Error Recovery for Pronunciation Dictionaries. In *Proc. Interspeech*, 2012.

- Ngoc Thang Vu, Florian Metze and Tanja Schultz. Multilingual Bottleneck Features and Its Application For Under-resourced Languages. In *Proc. SLTU*, 2012.
- Jochen Weiner, Ngoc Thang Vu, Dominic Telaar, Florian Metze, Tanja Schultz, Dau-Cheng Lyu, Eng-Siong Chng and Haizhou Li. Integration Of Language Identification Into A Recognition System For Spoken Conversations Containing Code-Switches. In *Proc. SLTU*, 2012.
- Tim Schlippe, Edy Guevara Komgang Djomgang, Ngoc Thang Vu, Sebastian Ochs and Tanja Schultz. Hausa Large Vocabulary Continuous Speech Recognition. In *Proc. SLTU*, 2012.
- Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, Haizhou Li. A First Speech Recognition System For Mandarin-English Code-Switch Conversational Speech. In *Proc. ICASSP*, 2012.
- Ngoc Thang Vu, Daniel Povey, and Tanja Schultz. Modeling Gender Dependency in the Subspace GMM Framework. In *Proc. ICASSP*, 2012.
- Daniel Povey, Mirko Hannemann, Gilles Boulianne, Lukas Burget, Arnab Ghoshal, Milos Janda, Martin Karafiat, Stefan Kombrink, Petr Motlicek, Yanmin Qian, Korbinian Riedhammer, Karel Vesely, Ngoc Thang Vu. Generating exact lattices in the WFST framework. In *Proc. ICASSP*, 2012.

2011

- Lori Lamel, Sandrine Courcinous, Julien Despres, Jean-Luc Gauvain, Yvan Josse, Kevin Kilgour, Florian Kraft, Le Viet Bac, Hermann Ney, Markus Nussbaum-Thom, Ilya Oparin, Tim Schlippe, Ralf Schlueter, Tanja Schultz, Thiago Fraga Da Silva, Sebastian Stueker, Martin Sundermeyer, Bianca Vieru, Ngoc Thang Vu, Alexander Waibel and Cecile Woehrling. Speech Recognition for Machine Translation in Quaero. In *Proc. IWSLT*, 2011.
- Ngoc Thang Vu, Franziska Kraus, and Tanja Schultz. Rapid building of an ASR system for Under-Resourced Languages based on Multilingual Unsupervised Training. In *Proc. Interspeech*, 2011.
- Ngoc Thang Vu, Franziska Kraus, and Tanja Schultz. Cross-language bootstrapping based on completely unsupervised training using multilingual A-stabil. In *Proc. ICASSP*, 2011.

2010

- Ngoc Thang Vu, Franziska Kraus, and Tanja Schultz. Multilingual A-stabil: A new confidence score for multilingual unsupervised training. In *Proc. SLT*, 2010.
- Ngoc Thang Vu, Tim Schlippe, Franziska Kraus, and Tanja Schultz. Rapid Bootstrapping of five Eastern European Languages using the Rapid Language Adaptation Toolkit. In *Proc. Interspeech*, 2010.
- Ngoc Thang Vu, and Tanja Schultz. Optimization On Vietnamese Large Vocabulary Speech Recognition. In *Proc. SLTU*, 2010.

2009

- Ngoc Thang Vu, and Tanja Schultz. Vietnamese Large Vocabulary Continuous Speech Recognition. In *Proc. ASRU*, 2009.

Own Student Theses

- Ngoc Thang Vu. Entwicklung eines vietnamesischen Spracherkennungssystems für große Vokabulare. *Diploma Thesis*, Institut für Anthropomatik, Fakultät für Informatik, University of Karlsruhe, 2009. *Supervision:* Prof. Dr.-Ing. Tanja Schultz

Co-advised Student Theses[1]

- Jing Sun. Prosodic Features for Code-Switching Speech Recognition. *Master Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2014, with Dominic Telaar
- Florian Deßloch. Spoken Term Detection using Deep Neural Networks. *Bachelor Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2014, with Dominic Telaar
- Qingyue He. RLAT Light - An Enhanced Version for Novices of the Rapid Language Adaptation Toolkit. *Master Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2014, with Tim Schlippe
- Heike Adel. Integration of Syntactic and Semantic Features into Statistical Code-Switching Language Models. *Master Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2014, with Dominic Telaar

[1]Unless otherwise indicated, all theses were completed under the primary supervision of Prof. Dr.-Ing. Tanja Schultz

- Marten Klose. Improving ASR for Non-native Speech Using Multilingual Data. *Bachelor's Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2013.
- Yuanfan Wang. Crosslingual Accent Adaptation: A Study Case for English and German with Chinese Accent. *Bachelor's Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2013
- Michael Ikkert. Implementierung und Evaluation eines Large Margin Estimation Algorithmus für HMMs. *Diploma Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2013, with Michael Wand
- Lukasz Gren. Unsupervised Language Model Adaptation for Automatic Speech Recognition of Broadcast News Using Web 2.0. *Diploma Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2013, with Tim Schlippe
- Dario Ernst. Bootstrapping Pronunciation Dictionaries with Multilingual Phoneme Recognition. *Bachelor's Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2013, with Tim Schlippe
- Jochen Weiner. Integrating Language ID into Code-Switch Speech Recognition. *Bachelor's Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2012, with Dominic Telaar and Florian Metze
- Wojtek Breiter. Haitian Creole Large Vocabulary Continuous Speech Recognition. *Student Research Thesis (Studienarbeit)*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2011, with Tim Schlippe
- Lukasz Gren. Enhancing Language Models for ASR using RSS Feeds. *Student Research Thesis (Studienarbeit)*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2011, with Tim Schlippe
- Edy Guevara Komgang Djomgang. Hausa Large Vocabulary Continuous Speech Recognition. *Student Research Thesis (Studienarbeit)*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2011, with Tim Schlippe
- Zlatka Mihaylova. Lexical and Acoustic Adaptation for Multiple Non-Native English Accents. *Diploma Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2011, with Tim Schlippe and Dominic Telaar
- Franziska Kraus. Cross-Language Bootstrapping based on completely Unsupervised Training. *Diploma Thesis*, Cognitive Systems Lab, Karlsruhe Institute of Technology, 2011