# Lexical and Acoustic Adaptation for Multiple Non-Native English Accents

Diploma thesis at the Cognitive Systems Lab (CSL)
Prof. Dr.-Ing. Tanja Schultz
Department of Computer Science
Karlsruhe Institute of Technology (KIT)

from

**Zlatka Mihaylova**

Advisors:
Prof. Dr.-Ing. Tanja Schultz
Dipl.-Inform. Tim Schlippe
Dipl.-Inform. Ngoc Thang Vu
Dipl.-Inform. Dominic Telaar

Begin: 4. November 2010
End: 4. Mai 2011

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

Karlsruhe, den 04. Mai 2011

**Abstract**

This work investigates the impact of non-native English accents on the performance of an large vocabulary continuous speech recognition (LVCSR) system. Based on the GlobalPhone corpus [1], a speech corpus was collected consisting of English sentences read by native speakers of Bulgarian, Chinese, German and Indian languages. To accommodate for non-native pronunciations, two directions are followed: Modification of the dictionary to better reflect the non-native pronunciations and adaptation of the acoustic models for native US English with non-native speech. The proposed methods for dictionary modification are data-driven. Therefore no language-specific rules are necessary: The idea is to extract a parallel corpus of phoneme sequences from phonetic transcriptions of native US English and accented English in the George Mason University (GMU) accented database [2]. With this corpus, Statistical Machine Translation models are generated to translate the US English pronunciations in the GlobalPhone dictionary into accented pronunciations which are then used as new pronunciation variants in the GlobalPhone dictionary. With the combination of the lexical and acoustic model approaches, relative improvements of 26.9% for Bulgarian, 33.2% for Chinese, 30.9% for German, and 53.2% for Indian accents are achieved.

**Zusammenfassung**

Diese Arbeit untersucht die Auswirkungen akzentbehafteter englischer Sprache hinsichtlich der Erkennungsqualität in LVCSR Systemen. Zu diesem Zweck wurde ein Sprachkorpus von verschiedenen Gruppen gesammelt, für die Englisch eine Fremdsprache darstellt: Muttersprachler für Bulgarisch, Deutsch, chinesische und indische Sprachen. Der akzentbehaftete Sprachkorpus basiert auf GlobalPhone [1] der Datenbank. Zur Verbesserung der Erkennungsqualität wurden zwei verschiedene Ansätze verfolgt: Modifikation des Wörterbuches um den Unterschieden in der Aussprache der verschiedenen Gruppen gerecht zu werden, sowie die Anpassung des akustischen Modells an die Nicht-Muttersprachler. Da die Methode der Anpassung des Wörterbuches datengestützt erfolgt, ist die Vorgabe von sprachspezifischen Regeln nicht notwendig. Stattdessen werden diese Regeln automatisch aus der akzentbehafteten Datenbank von GMU [2] ermittelt, in dem zusätzliche Aussprachevariante einer existierenden Sprachdatenbank hinzugefügt werden. Die Generierung neuer Aussprachevarianten erfolgt mit Hilfe von Statistischer Maschineller Übersetzung auf einem kleinen Korpus von IPA-basierten Transkriptionen. Durch die Kombination von Ansätzen, welche auf lexikalischen und akustischen Modellen basieren, konnte eine relative Steigerung der Erkennungsrate für die verschiedenen Muttersprachen erreicht werden: 26.9% für Bulgarisch, 30.9% für Deutsch, 33.2% für chinesische Sprachen und 53.2% für indische Sprachen.

**ACKNOWLEDGEMENTS**

We would like to thank the following persons: Prof. Dr. Tanja Schultz for being our supervisor at Cognitive Systems Lab (CSL) at Karlsruhe Institute of Technology (KIT), Dipl. Inf. Tim Schlippe, Dipl. Inf. Ngoc Thang Vu, Dipl. Inf. Dominic Telaar for their contribution with ideas, discussions and comments. Also thanks to Dr. Florian Metze, associate director of interACT lab at Carnegie Mellon University (CMU), Language Technology Institute (LTI) for his kind support.

Finally we would also like to thank all participants in the speech recordings for their time and effort.

# Contents

# 1. Introduction

## 1.1 Motivation

With advancing speech recognition technologies, the speech recognition performance achieves as low as 10% word error rates even for large vocabulary continuous speech recognition (LVCSR). However, recognizing low proficiency non-native speech is still a challenge with error rates two or three times higher than those for native speech [3]. Reasons for these are: In addition to the direct translations of grammar constructs from the first language, the non-native speakers also tend to pronounce the foreign language phonemes differently from the native speakers. It is a well known fact, that the earlier a child acquires a second language, the better it can produce the language-specific phonemes. Recent neuroscience research even shows that learning to discriminate between native and non-native sounds happens during the infant's first year [4]. As a consequence, people that acquired their second language at the age of 7 and later cannot discriminate the non-native sounds perfectly which results in distortion of non-native phonemes or phoneme sequences.

Many obstacles contribute to the low performance when recognizing foreign speech. Some of the speaker-related factors that have negative impact on speech recognition performance for non-native speech are [5]:

- High intra- and inter-speaker inconsistency 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.
- Higher cognitive load due to the non-nativeness.
- Reading errors in read speech.
- Slower reading with more pauses in read speech.
- Grammatically incorrect phrases in spontaneous speech.

Another general difficulty in building accent-specific automatic speech recognition (ASR) systems is the lack of accented speech data for training. There are few databases with accented speech and often the speakers have different mother tongues, which makes it difficult for the researchers to draw conclusions and compare the methods they experimented with. A widely used approach is to adapt a native ASR system to the non-native condition as it does not require the amount of data which is usually necessary to build an ASR system from scratch.

## 1.2 Goals

Since English is the language with a majority of non-native speakers (approximately 500 to 1,000 million second language speakers [6] [7]), this work is focused on investigating non-native English accents. The following work aims at:

- collecting corpus with accented speech
- investigating the impact of accent on ASR performance by comparing the accents with native speech or finding accent-specific rules
- analyzing accent consistency
- experimenting with Statistical Machine Translation (SMT) techniques to adjust the lexicon to better reflect accented pronunciations
- applying acoustic modeling techniques to improve the speech recognition performance for accented speech
- combining lexical and acoustic modeling to improve the speech recognition performance for accented speech

## 1.3 Structure

This work is structured as follows: in Chapter 2, the foundations of building ASR systems are briefly discussed. These are the most widely used speech recognition methods and algorithms as well as the approaches used to measure the quality of separate components or the performance of the whole ASR engine. In Chapter 3 related work to the topic of accented speech recognition is reviewed. In Chapter 4 the software tools used in this work are described. Chapter 5 summarizes the available speech data and offers statistics about the collected non-native accents and accented data. All corpora used for our data-driven approaches are described there. The experimental setup and the results from the lexical-and acoustic-based experiments are presented in Chapter 6. Finally, the conclusions of this work and future directions are defined in the last Chapter 7.

# 2. Background

## 2.1 Overview Automatic Speech Recognition

The fundamental problem of speech recognition is to find the most likely word sequence given audio file with speech. Factors like speaker variability, noisy environment and different properties of the recording equipment have negative influence on the recognition performance. The following equation (2.2) which is based on the Bayes' rule (2.1) summarizes the computational model used for large vocabulary continuous speech recognition (LVCSR):

$$\Pr(\mathbf{W}|\mathbf{X}) = \frac{\Pr(\mathbf{W})\Pr(\mathbf{X}|\mathbf{W})}{\Pr(\mathbf{X})} \tag{2.1}$$

$$\hat{\mathbf{W}} = \arg\max_{W} \Pr(\mathbf{W}|\mathbf{X}) = \arg\max_{W} \Pr(\mathbf{W})\Pr(\mathbf{X}|\mathbf{W}) \tag{2.2}$$

As a result of the digital signal processing, the acoustic signal is represented as a sequence of acoustic vectors $\mathbf{X} = X_1X_2...X_n$ that capture the short time spectrum of the speech 2.1.2. The goal is to estimate the most likely word sequence $\hat{\mathbf{W}} = \hat{W}_1\hat{W}_2..\hat{W}_m$ depending on the linguistic knowledge available for the language $\Pr(\mathbf{W})$ and the extracted acoustic rules. The probability of observing signal $\mathbf{X}$ given the fact that word sequence $\mathbf{W}$ is spoken forms the acoustic model $\Pr(\mathbf{X}|\mathbf{W})$. When computing the most probable word sequence, the denominator from the Bayes' rule $\Pr(\mathbf{X})$ is not considered since it does not play a role in the maximization of the function. Finally, to find the word sequence with the highest probability, a search strategy has to be applied. The most widespread search algorithms in speech recognition are A* and Viterbi search.

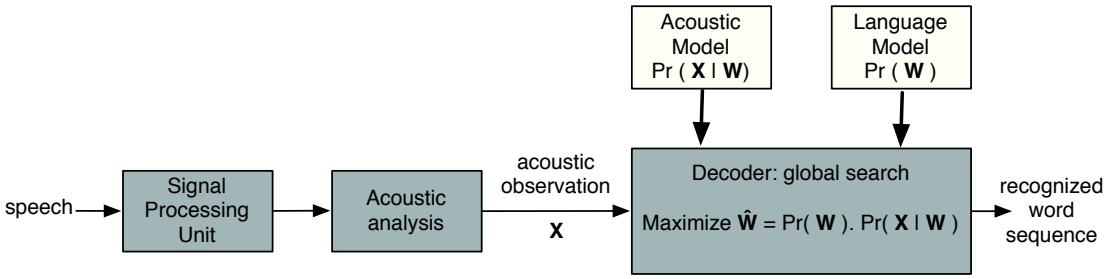


Figure 2.1: Computational model used for ASR.

To build an Automatic Speech Recognition (ASR) system, two steps have to be applied. In the first step, called training, a statistical model that represents the knowledge about the acoustic and linguistic properties of the language is built. To evaluate how good this model is, the model is tested with a smaller set of test speakers in the second step. Finding the most probable word sequence involves considering many parallel hypotheses and discarding those that seem to be improbable [8].This is known as decoding in the literature.

In Section 2.1.3, the approaches used for modeling the acoustic of speech are introduced. Details about how language information is embedded in a speech recognizer can be found in Section 2.1.4. The search algorithms used for finding the most likely word sequence are briefly discussed in Section 2.1.6.

### 2.1.1 Natural Language Structure

In general, language and speech are complex phenomena and can be analyzed on different levels [9]. Figure 2.2 shows the levels of language analysis.

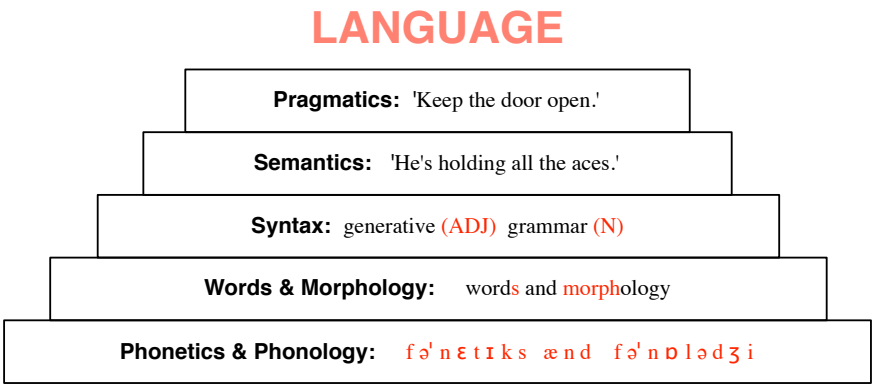


Figure 2.2: The levels of the language analysis

On the lowest level the speech is broken into small parts, called phonemes. The smallest sound that makes the difference between two words is called a phoneme. Phonemes sound different when put in a different context. The sound resulting from the different phonetic realization of a phoneme is called phone. The set of

used phones is something specific for a language and varies according to speaker and context.

The International Phonetic Alphabet (IPA) [10] offers a standard way of describing and categorizing the sounds produced when speaking. About 107 IPA symbols describe the basic consonants and vowels. These can be extended with the help of 31 diacritics that mark the place of articulation and 19 additional symbols for denoting sound properties such as length, tone, stress and intonation. IPA

Another way of describing the sounds primitives when speaking is the ARPAbet [11]. ARPAbet is a phonetic transcription alphabet developed specifically for sounds occurring in the General American English. The pronunciations of the Carnegie Mellon University (CMU) Pronouncing Dictionary (cmudict [12]) are given in ARPAbet. ARPAbet

Words are build from phonemes. A level higher, the language can be analyzed from the morphology's point of view. Morphology allows us to study the behavior of a word in the context of other words. Different languages have different morphological rules and therefore different degrees of morphological complexity.

On the higher levels, the language analysis deals with syntax, semantics and pragmatics. Although there is research on how syntactic, semantic and pragmatic knowledge can improve the speech recognition performance, these areas are from greater interest in fields like Artificial Intelligence and many branches of Linguistics. A way of including simple syntactic and semantic knowledge in the process of speech recognition is to use the word context information as done with the application of the Language Model (LM).

### 2.1.2 Signal Preprocessing

In order to be further analyzed, the speech signal must be converted into sequence of acoustic vectors. The acoustic vectors $\mathbf{X} = X_1X_2...X_n$ from equation 2.2 are calculated by dividing the speech signal into smaller blocks (typically $10ms$) and extracting the smoothed spectral estimates of these. 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). 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 are Linear Prediction (LP) coefficients and Perceptually weighted Linear Prediction (PLP) coefficients [13].

### 2.1.3 Acoustic Modeling

In the LVCSR where the word boundaries are not clear, the acoustic is modeled rather by using phones as fundamental speech units. The Hidden Markov Model (HMM) is the current most widely used representation of a phone. An HMM is a 5-tupple with the following elements [14]: HMM

- Set of states $S = \{S_1, S_2, .., S_N\}$. In any discrete moment, the system is in some of these states. In comparison to a Markov Model, the states in an HMM are hidden since the current state is unknown. Observing the system leads to an indirect conclusion in which particular state the system is in.

- A discrete alphabet $V = \{v_1, v_2, .., v_M\}$ of possible emissions.
- State transition probability distributions $A = \{a_{ij}\}$, where $a_{ij}$ is the probability of moving to state $S_j$ in the next step, if you are currently in $S_i$.
- The observation symbol probability distribution $\{B = b_j(k)\}$ which gives the probability of emitting symbol $v_k$ in state $S_j$.
- The initial state $\pi$ of the system which can be as well a number of states with initial probabilities.

The HMM that represents a phone consists typically of three emitting states for begin, middle and end of the phone. To form a word, the HMMs can be connected as shown in Figure 2.3. The probability of moving to state s_e, given current state s_m is $a_{me}$. The output of the system depends on the emitting probabilities. For example, the acoustic vector $y_3$ is generated from state s_m with probability $b_{y3}$.

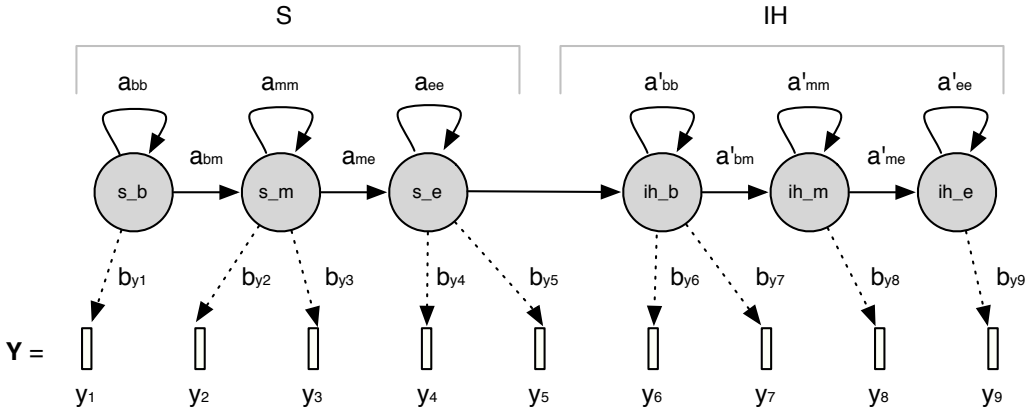


Figure 2.3: Connected HMMs building the word 'see'.

Generally, there are three ways for defining the emission probability functions $b$. When the emission probabilities of the HMMs occupy a discrete space, the HMMs are called discrete. In this case, there is a precomputed codebook of acoustic vectors organized as a lookup table. During training, the discrete emission probability function replaces the input acoustic vector with the closest vector from the codebook. This method guarantees faster training, but poor models, as the quantization leads to errors.

If the emission probabilities $b$ are probability density functions, the HMM is called continuous. Gaussian mixture models (GMMs) are used to represent the emission probability $b$ which makes this approach very accurate. The problem is, that it requires a larger set of training data since there are many parameters to be estimated. As pointed out in [8], in English, there are 45 phones used, but due to coarticulation there are many more sounds that occur when speaking English. If the left and right neighbors of a phone are considered, i.e. using triphones, much more than 45 HMMs must be trained. This will lead to models that are not robust if small amount of speech data is used for training.

The semi-continuous HMMs take advantage of parameter tying. They still use GMMs, but share those models when possible. For example, phones with different contexts share GMM with different state weights. This approach reduces the amount of parameters to be estimated enormously and offers a compromise between accuracy and trainability. However, to estimate which phones sound similar in different context, a binary phonetic decision tree is used.

As proposed in [15], binary phonetic decision trees are structures with question on each node that can be answered with "yes" or "no". The questions are related to the acoustic properties of the phone or its context. An example for a binary phonetic decision tree is shown on Figure 2.4. The end of the phone 't' ($t_e$ on the Figure) surrounded by 'o' from the left and 'uh' from the right (as in 'otter), sounds the same as 't' surrounded by 'e' and 'uh' (as in 'better'). Therefore, both phones are contained in the same leaf. The question that splits the clusters with lowest resulting entropy is placed on the top. The algorithm terminates when the training data for a phone or the information gain is small enough. The leafs contain phones which are acoustically indistinguishable despite of their different context.

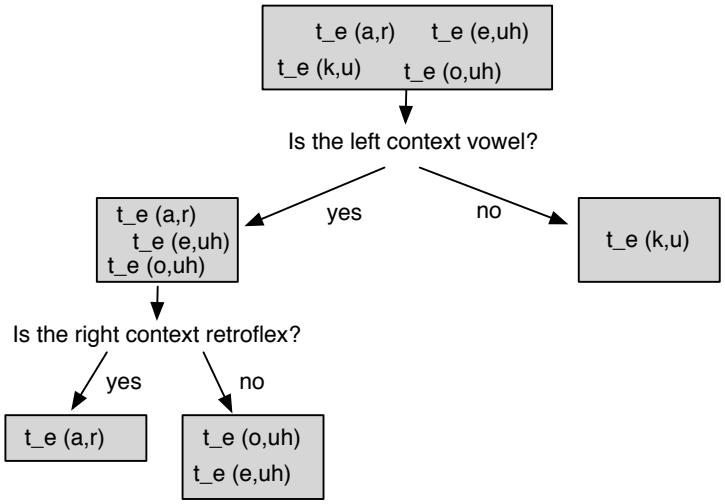


Figure 2.4: Example of a binary phone decision tree of the phone 't' in different context.

According to [14], there are three fundamental problems for the HMM design: evaluation, decoding and optimization. The optimization problem for the HMM design relates to the training phase. The idea is to adjust the parameters of the model, which maximize the probability of observing an acoustic sequence $Y$. Using the Forward-Backward algorithm, the acoustic sequence $Y$ is used as an input to train the model. The forward and backward variables needed for this algorithm can be computed with the Baum-Welch method.

The evaluation problem of HMMs answers the question how likely is it for a given observation to be the result from a given model. This problem is solvable with the Forward algorithm [14].

The decoding problem of HMMs needs to be solved in the decoding phase. Given an acoustic observation $Y$ and a model, the hidden state sequence has to be found.

Computing the hidden state sequence is equivalent to finding the spoken phones. There are effective algorithms for that. The simplest way of finding the hidden states is to compute all $Pr(S_i)$ at time $t$ and then choose the state with the highest probability. Another option that finds wider application in practice is the Viterbi algorithm. Viterbi aims to find the most likely state sequence for the duration of an utterance.

At this point information about the produced sequence of sounds is available. By adding the information from the pronunciation dictionary and the language model on the next step, a better guess (hypothesis) about what was spoken is produced.

### 2.1.4 Language Modeling

The language model (LM) used in speech recognition captures automatically extracted linguistic knowledge about the target language. An LM helps to select the best option for a word transition. Language and acoustic models are computed separately and then connected as illustrated in equation 2.2 to help the search algorithm to find the most likely word sequence.

The LM can be computed from a text corpus. It is a process of counting the occurrences of a given word $W$ in some context called history $H$. The history contains the previous $n-$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 with the help of equation 2.3.

$$Pr(W_1, ..W_k) = Pr(W_1).Pr(W_2|W_1).Pr(W_3|W_1, W_2)..Pr(W_k|W_1, ..W_{k-1}) \quad (2.3)$$

To compute the probability of a word to be seen in a defined context, a trigram LM is used as an example (equation 2.4). In the numerator, all the occurrences of the three words in the defined order ($W_{k-2}$, $W_{k-1}$, $W_k$) are counted and divided by the number the two words in the history appear in the training text.

$$Pr(W_k|H) = \frac{\#(W_{k-2}, W_{k-1}, W_k)}{\#(W_{k-2}, W_{k-1})} \quad (2.4)$$

For measuring the performance of an LM, a metric called perplexity is used. The perplexity measures how much new information a language model introduces to help the speech recognizer. The higher perplexity means that fewer new information was brought by the LM into the system which is not as helpful for the speech recognizer as a LM with a lower perplexity. An LM with lower perplexity produces in many cases a greater recognition accuracy. From a technical point of view, the perplexity is a entropy-based measurement, which computes the geometric mean of the branching factors of a language or grammar. The perplexity is computed from equation 2.5, where $H(W)$ is the cross-entropy of word sequence $W$.

$$PP = 2^{H(W)} \quad (2.5)$$

The quality of the n-gram LMs depends on the amount and the quality of the data used for training the model. However, it is impossible to find training data that covers all the possible word combinations of the language. To escape the problem of assigning a zero probability to a phrase that actually can occur as valid language construct but does not occur in the training text different LM smoothing techniques are used:

The strategies used to implement LM smoothing are discounting, backing-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. Different smoothing techniques and an example of a discounting algorithm, called "Good-Turing Estimate" are reviewed in [16].

Another way to smooth the probability distributions of the n-grams is to back off to lower order models. If given n-gram does not occur in the training data, usually the n-1-gram distributions are used to calculate the new distribution. However, if the count of a given n-gram is greater than zero, then the new distribution can be calculated according to different back-off smoothing strategies. A comparison of the most widely used smoothing strategies depending on the training data is proposed in [16]. They prove that the algorithm proposed by Katz performs better than Jelinek-Mercer when applied on a bigger training set. The performance of the Jelinek-Mercer method is better on small amounts of training data. They conclude that the modified Kneser-Ney smoothing outperforms both methods consistently.

Another way of smoothing an LM is via linear interpolation. No matter if the n-gram is already seen in the training data or not, the new probability distribution of the interpolated n-gram is recursively calculated with the following formula:

$$Pr_{interp}(W_k|W_{k-n+1}^{k-1}) = \lambda \hat{Pr}(W_k|W_{k-n+1}^{k-1}) + (1-\lambda) Pr_{interp}(W_k|W_{k-n+2}^{k-1}) \quad (2.6)$$

where $\hat{Pr}(W_k|W_{k-n+1}^{k-1})$ is the maximum likelihood estimate of the n-gram. The weight $\lambda$ of the interpolation is derived empirically from a development set.

### 2.1.5 Pronunciation Dictionary

The pronunciation dictionary is the connection between words and phones. The pronunciation dictionary used for training LVCSR systems is a mapping between words and their pronunciations used in the training set. Different pronunciation dictionaries can be used for training and decoding. Pronunciation dictionaries are referred to with the term "dictionary" only or substituted with the term "lexicon". There is no standard for how to format a dictionary. Depending on the system's structure, the format can vary. Usually it is a map-like structure with words as keys and pronunciation strings as values.

Since a word can have multiple pronunciations, the dictionary structure should allow adding pronunciation variants and a way of marking them as pronunciation variants of the same word.

The keys (words) of the dictionary are language-dependent and usually use the alphabet of the language to be recognized. The pronunciations of the words is given

as phonetic transcription defining how the word is spelled. There are standard phonetic transcriptions systems like IPA, ARPAbet and many others developed for general or special purposes. IPA and ARPAbet which are used for the experiments are discussed in Section 2.1.1.

Dictionary entries of English words and their transcriptions in ARPAbet look as follows:

```
efficient  IH F IH SH AX N T
efficient(2)  IX F IH SH AX N T
efficiently  IH F IH SH AX N T L IY
```

Depending on the application, the size of the dictionary can vary from a few to millions of words. Speech recognizers for LVCSR usually use thousand words and above.

### 2.1.6 Search Algorithms for ASR

The last step of the speech recognition is to find the most likely word sequence $\hat{\mathbf{W}}$ from equation 2.2. Generally, decoding is a search problem. Therefore as a last step in the decoding process the choice of search algorithm and the method for search space construction are very important.

As objective function for the search, the inverse of the Bayes' posterior probability is defined:

$$C(\mathbf{W}|\mathbf{X}) = -\log(\Pr(\mathbf{W})\Pr(\mathbf{X}|\mathbf{W})) \tag{2.7}$$

The log probability is applied to substitute the multiplication with addition to make the calculations easier and more robust. The problem of finding the word sequence $\hat{\mathbf{W}}$ with maximal probability is transformed into the problem of finding the search path with the minimal cost:

$$\hat{\mathbf{W}} = \arg\min_{W} C(\mathbf{W}|\mathbf{X}) \tag{2.8}$$

Commonly employed algorithms are A* and Viterbi search as well as many specialized versions of those (e.g. implementation of A* with Stack [17], Viterbi search with beam pruning [18]). The arguments for using Viterbi search are connected with the time-synchronous nature of the speech signals. The advantage which the A* algorithm offers is a good performance in larger search spaces with a higher branching factor.

Viterbi search

Generally, Viterbi search is an uninformed breadth-first search technique. For the purposes of LVCSR, the time-synchronous version of the algorithm[1] is used. To keep the required information to backtrace the best path, a backtracking pointer is used in the implementation of the algorithm. Another improvement that saves time by

[1] The time-synchronous Viterbi search takes advantage of the dynamic programming principle and achieves efficiency by dynamically generating the search space.

reducing the search space is the usage of a beam that throws away the paths with high costs. For example, a threshold $T$ is defined and on every step $t$, the most promising nodes in the next step $t + 1$ are expanded. All branches with a score greater than the cost for reaching the best candidate plus the threshold value are pruned.

If the beam width is too small, the Viterbi algorithm can miss the optimal path. Therefore this algorithm will not guarantee optimality depending on the beam width. Wrong beam parameters may result in performance degradation since the correct hypothesis may be pruned.

A* search

Another search technique used for speech recognition tasks is A*. The nature of A* is a depth-first informed search algorithm. The disadvantage of A* against Viterbi in the context of speech recognition consists in the necessity of defining a heuristic function $h$. Since it is a depth-first search, A* is usually implemented as a Last In First Out (LIFO, also stack) data structure, e.g. the last node in the stack is the next to be further expanded.

A* is an algorithm that guarantees optimality if the heuristic function never overestimates the true distance to the goal [19]. This means that the heuristic is an optimistic estimation of the complete path to the goal and the true cost of this path will be always higher than that. If there is a path with lower cost, then the algorithm guarantees that all of the nodes on this path will be expanded. The stack-based implementation of A* (stack-decoder) also guarantees to find the global optimum.

The search space can be represented as a graph. Since it is usually very large, some techniques for accelerating the search performance must be used. There are two widely employed techniques for faster search algorithms: sharing and pruning.

Key step in the construction of the search space is building a lexical tree from the dictionary. The lexical tree is built from subword parts (phonemes) with the help of the pronunciations defined in the dictionary. Words that contain the same prefixes share arcs in the tree. For example, in Figure 2.5, the words 'two' and 'to' are allophones. Although they share all phonemes, they are still represented as different words.

The deficiency of this model is that LM probabilities can be used after the word is known (once a word-node is reached). This can be helped with the LM probabilities distributed among the subword parts rather than only on path ends. The lexical trees that assign probabilities on internal nodes are called "factored"[20].

If an unigram LM is used, the structure in Figure 2.5 works well. But for higher order LMs, things are more complicated since multiple copies of lexical trees have to be stored in the memory. To handle the higher complexity of the search space, three possibilities are listed:

- Static decoder with precomputed search network containing LM, dictionary and AM information (Finite State Transducer)
- Dynamic generation of the n-gram search space via graph search algorithms

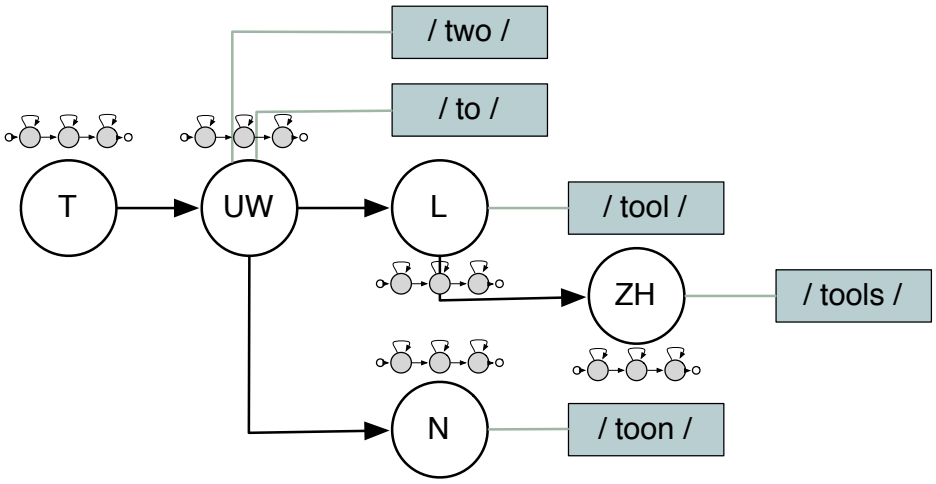


Figure 2.5: Example of sharing subword parts (phonemes). The internal nodes represent phonemes, the leafs contain words.

- Usage of a multipass framework that roughly explores the space to reduce it first and then performs detailed analyses to find the most promising word sequence.

A multipass decoder performs speech recognition in two steps: Less detailed models[2] are used as starting point in the first pass to generate a list with the $N$-best hypothesis. To further simplify the process, a lattice as in Figure 2.6 is generated. The lattice is a graph with words on the arcs connected in a time synchronous manner that represents the alternative hypotheses of the speech recognizer. The last step is by using computationally more expensive but also more accurate techniques to find the best path in the lattice.

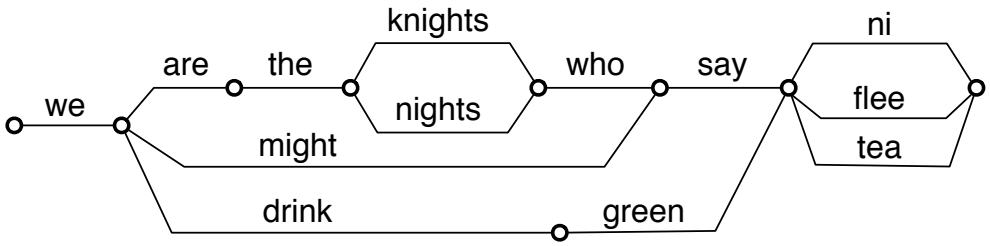


Figure 2.6: Example of a lattice with alternative hypotheses.

Forward-Backward search

The first pass (also called "forward" search) of a multipass decoder is fast and computes the forward scores for the next few frames. A time synchronous Viterbi search is usually the algorithm of choice for this step. Then in the second pass (known as "backward" search), the forward scores are used as the heuristic function necessary for performing informed search. The second pass runs backwards, i.e. from the last frame to the first frame and is implemented as a stack managed with the A*

[2] low-order n-grams, context independent phoneme models

strategy. The result of this "forward-backward" search is a list with $N$-best word sequences which are then transformed to a lattice.

### 2.1.7 Acoustic Model Adaptation

Adapting an already existing system to better handle new conditions succeeds on different levels. Either the LM can be adapted by using text corpora that are closer to the data used for decoding or the acoustic model. Since in the following experiments only read speech is recognized, the LM is fixed. Therefore the topic of LM adaptation is not further discussed. The focus is on adapting the acoustic model.

Speaker 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 dictation software to improve the speech recognition performance for the user of the software which means transforming the speaker-independent system to a speaker-dependent one. As shown in Figure 2.7, if not enough data to train a real speaker dependent system is present, general models can be used as a starting point. The idea of speaker adaptation is using a small amount of specific speech data to calibrate the already trained general models for the new conditions. Adaptation is a powerful concept based on comparable methods which humans use to understand speech with never seen properties.

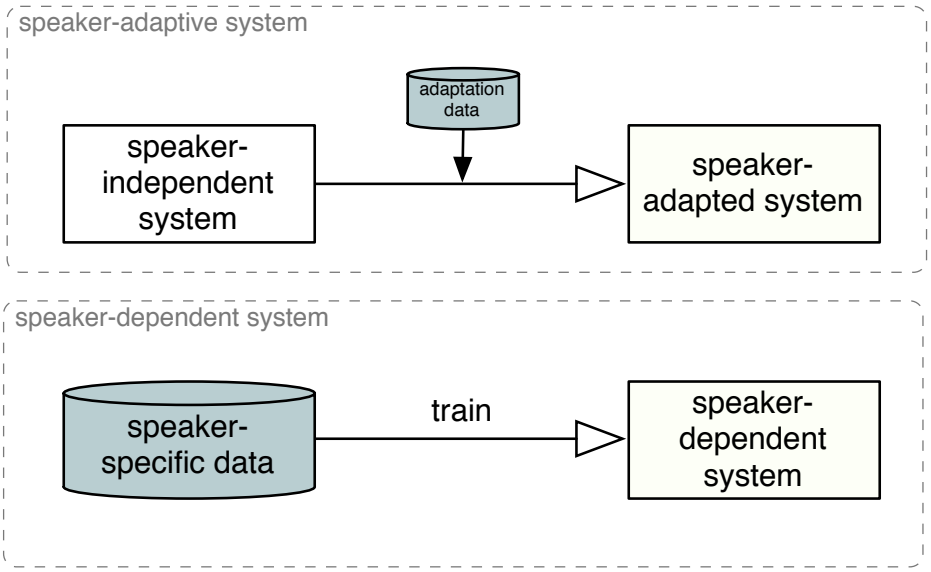


Figure 2.7: Comparison of speaker-adapted and speaker-dependent system.

There are different kinds of speaker adaptation. For example batch adaptation which is adapting the system only once. Another possibility is incremental adaptation which runs the adaptation process in the background and adapts while the user is speaking. Generally, the adaptation can also be categorized according to the available transcriptions as supervised or unsupervised adaptation.

There are two adaptation techniques that proved to be very successful for recognition of accented speech: Maximum a Posteriori (MAP) and Maximum Likelihood

Linear Regression (MLLR). Experiments show that MAP adaptation outperforms MLLR when bigger amount adaptation data is available. The reason is that MAP adaptation collects statistics and then adjusts the parameters only of the distributions that are observed in the calibration data. In contrast, the MLLR approach defines a model that shifts all general distribution parameters towards the observed data according to some pre-calculated model. The following sections offer a detailed view of both adaptation methods.

MLLR

### Maximum Likelihood Linear Regression (MLLR)

Maximum Likelihood Linear Regression (MLLR) is a method that transforms the parameters of the emission Gaussian density functions of the HMM in a linear manner. This kind of transformation captures linear relationships between the general models and the adaptation data. As discussed in [21], the transformation can be applied either in the model space or in the feature space, but for speech recognition tasks, a transformation in the model space has proven to be more appropriate.

When using MLLR adaptation, either exclusively the means or additionally the variances of the Gaussian distributions are transformed. It is also possible to decouple the means from the variances and transform them separately which is defined as unconstrained MLLR in [21].

Generally, the transformation of the mean $\mu$ can be done with the following equation:

$$\hat{\mu} = \mathbf{A}\mu + \mathbf{b} \tag{2.9}$$

where $\mathbf{A}$ is the regression matrix and $\mathbf{b}$ is the additive bias vector for a phone class. These two parameters are estimated through the Expectation Maximization (EM) algorithm [22]. The goal is to optimize the $\mathcal{Q}$-function given new data and already trained models:

$$\begin{aligned} &\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \\ &= K - \frac{1}{2}\sum_{m=1}^{M}\sum_{\tau=1}^{T} p(q_m(\tau)|\mathcal{M}, \mathbf{O_T})[K^{(m)} + \log(|\hat{\mathbf{\Sigma}}^{(\mathbf{m})}|) + \\ &\qquad (\mathbf{o}(\tau) - \hat{\mu}^{(m)})^T \hat{\mathbf{\Sigma}}^{(m)-1}(\mathbf{o}(\tau) - \hat{\mu}^{(m)})] \end{aligned} \tag{2.10}$$

where $q_m(\tau)$ is the Gaussian $m$ at time $\tau$, $\mathbf{o}(\tau)$ is the adaptation observation at time $\tau$, $K$ is constant derived from the transition probabilities, $\mathcal{M}$ contains the parameters for the trained models and $\mathbf{\Sigma}$ is the estimated variance matrix.

If constrained MLLR is applied, the transformation of the covariance matrices uses the same parameters as for the transformation of the means:

$$\hat{\mathbf{\Sigma}} = \mathbf{A}\mathbf{\Sigma}\mathbf{A}^{\mathbf{T}} \tag{2.11}$$

In the unconstrained case, the regression matrix is estimated separately.

The adaptation of the means results in bigger WER improvements than the adaptation of the variances.

As mentioned previously, the MLLR adaptation leads to better results than MAP if only a small dataset of transcribed speech is available because the transformation can be applied globally and thus compensates for the missing information.

#### Maximum a Posteriori (MAP)

MAP

The Maximum a Posteriori (MAP) adaptation tries to reestimate the HMM parameters given an observed signal using:

$$\mathbf{\Theta_{MAP}} = \arg\max_{\Theta} p(\mathbf{X}|\mathbf{\Theta})p(\mathbf{\Theta}) \tag{2.12}$$

with $\mathbf{\Theta}$ as the HMM parameter vector, $p(\mathbf{X}|\mathbf{\Theta})$ the probability density function (pdf) of the observed data given the model parameters and $p(\mathbf{\Theta})$ the prior pdf of $\mathbf{\Theta}$. This equation is derived from the Bayes' rule to find a new HMM parameter vector by maximizing its probability given calibration data.

If no information about $p(\mathbf{\Theta})$ is present, the MAP estimator is the same as Maximum Likelihood Estimator (MLE) and the term $p(\mathbf{\Theta})$ is dropped from equation 2.12 since it is a constant, i.e.:

$$\mathbf{\Theta_{MLE}} = \arg\max_{\Theta} p(\mathbf{X}|\mathbf{\Theta}) \tag{2.13}$$

If adaptation data is present, the prior probability can be modeled as a Gaussian distribution instead of the uniform distribution as assumed by MLE. So the difference between MAP and MLE is how the prior function is assumed.

As discussed in [23], the first step for a MAP adaptation is to find the appropriate prior distribution. Again, this can be done with the EM algorithm by iteratively optimizing the auxilliary function shown in equation 2.10.

### 2.1.8 Measuring 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 a reference text which is the true transcription of what is said, gives 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:

WER

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

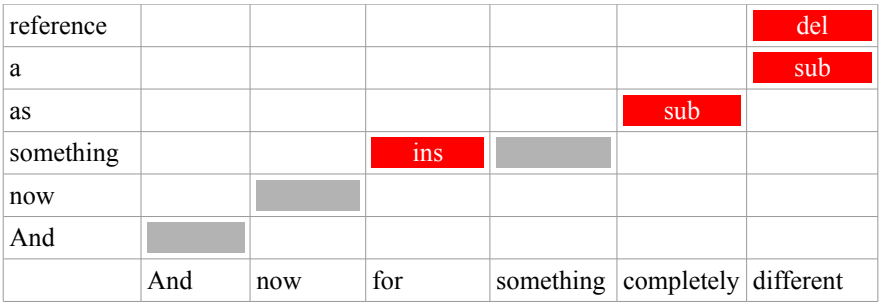


Figure 2.8: Example of the errors that an ASR system can make. On the vertical axis is the reference and horizontally the output sequence of the recognizer.

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

$$WER[\%] = \frac{\#substitutions + \#deletions + \#insertions}{\#words(reference)} * 100 \tag{2.14}$$

The equation above shows that the WER can exceed 100%, especially in the case when the speech recognizers tends to insert words.

WA The performance of an ASR system can be also measured as word accuracy (WA):

$$WA[\%] = (1 - WER) * 100 \tag{2.15}$$

SMT

## 2.2 Statistical Machine Translation

### 2.2.1 SMT Introduction

The problem of statistical machine translation (SMT) is similar to the ASR problem [24]. If the task is to translate a German sentence $g$ to the English sentence $e$, the problem can be expressed as follows:

$$\hat{e} = \arg\max_e \Pr(e|g) = \arg\max_e \Pr(e)\Pr(g|e) \tag{2.16}$$

In equation 2.16, to compute the most likely English translation of the German sentence $g$, two probabilities $\Pr(e)$ and $\Pr(g|e)$ are computed. $\Pr(e)$ represents the LM[3]. To estimate its probability, the same approaches as described in Chapter 2.1.4 are applied. For estimating $\Pr(e)$ an English text corpus is needed. In SMT, the TM second term $\Pr(g|e)$ is called a translation model (TM). It represents the likelihood of translating a sentence $e$ to a sentence $g$. To compute this model, a bilingual parallel corpus is required. To create a parallel corpus with English and German sentences, the English words are aligned to the German words. This is not a trivial task for the following reasons: more words can be translated with one or the words from the source and target corpora can be positioned differently.

Although the fundamental equation for SMT looks a lot like the fundamental equation for ASR, both fields have differences. Sometimes the mapping of words in the

[3]LM of English language in this case

source and target languages is not perfect. For example, a single source word can be translated to $0..n$ words from the target language. Consider the following mapping:

**Brotmesser ↔ bread knife**

Apparently, the translation of the German word "Brotmesser" results in two words in the target language. This relationship can be modeled with the "fertility" function ($n$-function) as suggested in [25]. More precisely, the $n$-function represents the probability of producing $n$ English words, given a specific German word. Thus, a probability $n(2|Brotmesser)$ (with $\phi = 2$) shows how likely it is to translate "Brotmesser" with two English words.

Another difference between the general computational models of ASR and SMT is the fact, that the sequence of the words can be different for the source and target languages. The following sentences serve as an example:

**Dir kann ich alles sagen. ↔ I can tell you everything.**

Although a perfect match for all words is present, the words from the target language do not have the same word order as in the source language. This permutation of the words is known in the literature [25] as distortion function ($d$-function). The simplified distortion function can be interpreted as follows: $d(5|4)$ defines how likely it is for the fourth word in the source sentence (e.g. "alles") to be translated to the fifth word from the target sentence ("everything"). Having only the example sentence above in the corpus, there is a 100% probability.

### 2.2.2 Measuring SMT Performance

Since in Machine Translation a sentence can have many correct translations, there are many possibilities to measure the quality of MT systems. The most popular ones are Bilingual Evaluation Understudy (BLUE) [26], NIST [27] and Metric for Evaluation of Translation with Explicit ORdering (METEOR) [28]. The standard WER common for evaluating ASR systems is also used. However, as the word order in translations is more flexible, a candidate translation can have high WER, but still be a good translation. Therefore WER is not appropriate for evaluating MT systems.

TER

An automatic metric for MT quality is the Translation Error Rate (TER). It is based on the number of edits needed to change the candidate sentence into the reference sentence.

BLEU

The Bilingual Evaluation Understudy (BLEU) is the most commonly used evaluation metric for SMT [26]. To evaluate the quality of an SMT system with BLEU scores, a candidate translations with a set of good quality human-made references for each sentence is required. BLEU is based on computing the weighted average of all modified n-gram precisions on the reference set. A modified n-gram precision is counting the maximum occurrences of each n-gram from the candidate sentence in the references, divided by the overall number of n-grams in the candidate translation. The BLEU score for each candidate is between 0 and 1. For a complete corpus it is the arithmetic mean of all sentences' BLEU scores.

NIST

Another metric is the modified BLEU metric known as NIST (named to the US National Institute of Standards and Technology (NIST)). The exact description of

NIST MT metric as well as experimental results on its performance can be found in [27].

METEOR

METEOR is a MT metric based on the harmonic mean of unigram precision and recall. To get the METEOR score, first example alignments between candidate and reference sentence are generated. One of these alignments is then selected according to the rules defined in [28]. The unigram precision P and the unigram recall R are then combined using equation:

$$F_{mean} = \frac{10PR}{R + 9P} \tag{2.17}$$

The recall is weighted higher than the precision. Since only unigrams are taken into account, an additional penalty is introduced to incorporate the n-gram information. The final score is computed according to the equation:

$$Score = F_{mean} * (1 - Penalty) \tag{2.18}$$

For more information regarding the METEOR metric, please refer to [28].

For tuning of an SMT system, the BLEU score is used as an optimization criterion for Minimum Error Rate Training (MERT) 2.2.3 further in this work.

MERT

### 2.2.3 Minimum Error Rate Training (MERT)

There are different methods to optimize SMT model parameters and improve the machine translation quality evaluated with multi-reference WER or Bilingual Evaluation Understudy (BLEU) score. Some of these methods are evaluated in [29]. The weights of the different SMT parameters can be estimated with Minimum Error Rate Training (MERT). Given a corpus with representative sentences and their translations, for every pair of sentence and its reference, the minimum of the sum of all errors is to be found. The criterion for finding the minimum is an objective function $E(reference, sentence)$ that compares the both sentences. This procedure generates new weights to be used in combination with the SMT parameters (fertility, distortion, LM).

# 3. Related Work

Over the past years, a lot of research in the area of recognizing accented speech has been done. Many strategies to improve the recognition performance have been applied. Generally, these methods can be classified as data-driven or rule-based. As the name suggests, the data-driven approaches require significant amounts of speech data or sometimes other text-based data, such as detailed phonetical transcriptions as in this work. These resources are used in many ways, e.g. to find the differences between native and non-native pronunciation, to recompute the polyphone decision tree or to retrain or adapt the models to the new data. In comparison to data-driven methods, the rule-based approaches cannot be automized since they rely on rules derived from linguistic knowledge.

Since the accented speech data is the basis for all data-driven approaches, available accented speech databases are described in Section 3.1.

Another classification of the methods to deal with accented speech is proposed in [30]. Basically, the straight forward approach to build a speech recognition engine for non-native speech of some target language is to modify some existing ASR engine for this language. According to the modified part of the ASR system, the following categories of methods are introduced:

- Lexical modeling
- Acoustic model adaptation

In Sections 3.2 and 3.3, the related work that has been done in each of these categories is introduced.

## 3.1 Accented Speech Corpora

The majority of available accented corpora target the English language. However, native accents (often referred to as "dialects" of a language) differ from non-native accent in many aspects: Fluency, consistent pronunciations, dialect words which may

be completely different from the non-dialect word, etc. Two well-known corpora with native English Accents are for example TIMIT [31], IViE (Intonational Variation in English) [32] databases. As this work is focused on non-native accents, only foreign-accented speech corpora are discussed further.

A detailed overview of the existing non-native speech databases is given in [33]. Some of the corpora related to this work are listed here:

- Read non-native English:
  - **CSLU** [34]: Foreign Accented English, approximately 30 hours, 4,925 telephone quality utterances from native speakers of 23 languages: Arabic, Brazilian, Portuguese, Cantonese, Czech, Farsi, French, German, Hindi, Hungarian, Indonesian, Italian, Japanese, Korean, Mandarin, Malay, Polish, Iberian, Portuguese, Russian, Swedish, Spanish, Swahili, Tamil, Vietnamese.
  - **ISLE** [35]: German and Italian accented English, 18 hours, 46 speakers, 1,300 words of an autobiographic text and 1,100 words in short utterances per speaker.
  - **ATR-Gruhn** [36]: 89 non-native speakers of English with origins in China, France, Germany, Indonesia and Japan, proficiency ratings of the speakers performed by native English speakers with teaching experience, per speaker: 25 credit card number sequences, 48 phonetically rich sentences and six hotel reservation dialogs.
  - **MC-WSJ-AV** [37]: The Multi-Channel Wall Street Journal Audio Visual Corpus, 45 speakers with varying native and non-native accents, captured using microphone arrays, as well as close-up and wide angle cameras.
- Spontaneous non-native English:
  - **Wildcat** [38]: 24 native and 52 foreign-accented English speakers: Chinese (20), Hindi/Marathi (1), Italian (1), Japanese (1), Korean (20), Macedonian (1), Persian (1), Russian (1), Spanish (2), Telegu (1),Thai (1), Turkish (2), scripted and spontaneous speech recordings, goal - to measure the communicative efficiency of native, non-native and mixed dialogues.
  - **AMI** [39]: English from native and accented English of Dutch people or other. 100 hours of spontaneous meetings recorded with close talk microphones.
  - **Hispanic-English corpus** [40]: 11 speaker pairs, 20 hours, spontaneous conversational speech, 2200 Spanish and English sentences read speech (50 Spanish, 50 English sentences per speaker), English sentences selected from the TIMIT database.

The CSLU Foreign Accented English corpus differs from the corpus collected in this work in that it contains only telephone-quality recordings of accented speech. The ISLE database contains German-accented English, but it is composed of words and

short utterances taken from English textbooks. ISLE is specifically developed to support applications for foreigners learning English. The ATR-Gruhn speech corpus covers multiple non-native accents with German and Chinese among them. However, the read sentences are not in the broadcast news domain.

## 3.2 Lexical Modeling for Accented Speech

The lexical modeling is the task of modifying the decoding dictionary so that it reflects the pronunciation differences between native and non-native speech. To generate pronunciation variants, different approaches can be applied.

In [41] both data-driven and rule-based methods are explored to analyze Japanese-accented English. For the automatic generation of pronunciation variants, a phoneme recognizer is used to find the most frequent confusion pairs of native and non-native phones. A second experiment suggests to add variants to the pronunciation dictionary derived from linguistic knowledge. Another way of adapting the lexicon to the non-native condition is by adding words with English origin but Japanese pronunciation. Such words are common in Japanese and are consistently pronounced. A next experiment involves manually creating a dictionary with the most common pronunciations of the 100 most frequent words. This method results in the best recognition performance improvement. The automatically generated pronunciations also proved to be helpful. However, the improvement by adding the linguistically-motivated variants to the lexicon is slightly worse compared to the lexicon containing automatically generated and hand-coded variants.

In [42], the authors propose another data-driven method for automatically generating pronunciation variants for accented speech. The approach does not require accented speech but fully relies on two speech databases with native speech in the source and target languages. Their method is designed for usage with isolated words. They extend the pronunciation dictionary with non-native pronunciations containing 375 short commands. All experiments are conducted with English-accented German and German-accented French. They input the German pronunciations into the phoneme recognizer trained with native British English so they get English-accented phoneme transcriptions of German words. In the next step, a decision tree trained to map the German native pronunciations and English accented variants is built. To build an extended dictionary, the generated accented pronunciations are then added to the canonical German pronunciations. By only using the new extended dictionary, they achieved an absolute improvement of 1.49% against the baseline and 5.25% when applied together with MLLR adaptation.

A common approach for generating variants that better reflect the underlying pronunciation is to run a phoneme recognizer on the target speech and then align the result with an existing pronunciation. This way the generated alignment is used to extract rules that are further applied to ”translate” the existing lexicon into a new one. The difficulty in this approach is to filter the systematic phoneme errors (substitutions, deletions, insertions) from the random errors (false segmentation, recognizer transcription errors, etc.). In [43] native British English accents are investigated. The accented data is transcribed at phoneme level and aligned to already existing lexicon entries from the source accent. The comparison of both results in context-dependent phoneme replacement rules. These rules are further clustered

in a decision tree by maximizing the tree "purity"[1]. The work is extended in [44] by deriving an American English pronunciation dictionary from the initial British English lexicon.

Another question when relying on the phoneme recognizer output is how to align the generated transcription with the dictionary pronunciation entry. [45] proposed a measure that incorporates the association strength between phones. The association strength is a measure based on statistical co-occurrences of phones. The method tries to differentiate between the dependent and independent phone co-occurrences by examining the independence assumption of the conditional probabilities of phones' associations.

## 3.3 Acoustic Modeling for Accented Speech

The authors of [5] explore different strategies for adapting native AMs with data from non-native speakers. There is an investigation if adapting with the native speech of the non-native speaker will achieve more gains than the non-native speech. It came out that the performance of the system adapted with the native data is worse than that of the baseline system. In contrast, using non-native speech from a specific accent gave significant improvement. The best performance reported in this paper is obtained by running two additional forward-backward iterations with accented data on the baseline system and then applying linear AM interpolation with empirically calculated weight factors.

As the native and non-native acoustic realizations differ from each other, it is proven that modifying the acoustic models results in greater WER improvement compared to lexical modeling methods. For example, retraining the native models with included non-native data lowers the WER by approximately 20% absolute as reported in [41]. Since non-native speech is usually more difficult to find, different experiments with interpolating the native and non-native AMs result also in better WER than the baseline system. For example, the interpolated AM can be calculated by shifting the means and variances between the native and the non-native models, i.e. finding empirically the optimal weight coefficient $w$ on a development set and applying:

$$\mu_{ij}^{new} = w\mu_{ij}^{accent} + (1-w)\mu_{ij}^{native} \tag{3.1}$$

where $\mu_{ij}$ is the $j$-th component of the mean vector for the $i$-th phoneme or senone[2]. The procedure is similar for the variance matrices. According to [41], this method resulted in a WER reduction from 67.3% to 45.1%.

Another group of researchers [40] focuses on Hispanic speakers of English. First, they developed a protocol for collecting spontaneous speech. Then some initial experiments using the collected data are conducted. Their strategy for spontaneous speech collection is based on a game-like approach and the subjects are recorded in pairs. For the experiments, a conversational recognizer trained with the SWITCHBOARD corpus with telephone channel quality is used. A vocabulary of 3,335 words

[1] Tree purity is a measure defined in the referenced paper.

[2] A senone is an atomic acoustic unit often used in speech recognition.

covering all the train and test utterances was compiled. Each pair was separated by using one of the speakers in the training set and the other in the test set. They report an initial WER of 73.6%, which could be reduced to 66.6% after MLLR speaker adaptation.

Automatic speech recognition for different native English accents is the subject of [46]. The authors focus on broadcast news speech data from the following geographic regions where English is an official language: US, Great Britain, Australia, North Africa, Middle East and India. They have a baseline system trained with 180 hours of broadcast news (US). To build an accent-independent AM, they pool part of the US English corpus together with the whole amount of accented data. This accent-independent system delivers better results for all accents, even for the US test data. The accent-independent system is further adapted to the specific accents by MAP speaker adaptation. The adaptation achieves improvement for the Middle East, Indian, US and Great Britain speech data, but does not have influence on the Australian test set and results in an worse performance on North African English. The final goal is to classify the accent of the input speech and recognize what was said by loading the proper accent-adapted and gender-adapted model. The group achieves an average of 19.18% WER for the final configuration consisting of accent identification and speech recognition.

A new approach based on a phone confusion matrix is proposed in [47]. The target is the French accent in a small vocabulary of 314 English words. To calculate the phone confusion matrix, the phone recognizers trained with native French and non-native English speech are used. The output of both phoneme recognizers when confronted with accented utterances are first (native) and second (non-native) language phone sequences. These phone sequences are then force-aligned and a confusion matrix is extracted from the alignment. Finally, the HMM of every phoneme is modified by adding alternative paths for each confusion pair.

In [48] the following goal is aimed: A system that optimizes the performance of four different non-native English accents without degrading the recognition performance of the native language. This goal is achieved by using Multilingual Weighted Codebooks. The best result is a 4.4% absolute word accuracy improvement and the experiments are conducted with non-native English with Spanish, French, Italian and Greek accents.

Different techniques that improve the recognition performance for non-native speech are compared in [49]. The study uses spontaneous German-accented English and investigates the following approaches: A bilingual model, a model built from mixed (native and non-native) speech, MAP speaker adaptation, AM interpolation, and Polyphone Decision Tree Specialization (PDTS). The results show that an AM trained with pooled native and non-native speech gives small improvement due to the greater variety of the new pool. Since the outcome of this approach depends on the proportion of native to non-native speech, an appropriate weighting of the native and non-native AMs reduces the errors by more than 10% absolute. Comparable results can be achieved with MAP speaker adaptation. The lowest WER was achieved by applying PDTS followed by MAP adaptation.

PDTS [50] is a technique developed for adapting multilingual LVCSR systems to new languages with limited amounts of adaptation data. The PDTS technique is

an Acoustic Modeling technique complementing the common AM approaches (e.g speaker adaptation). In [51], it is successfully used to adapt a native English baseline system to German-accented English. Combined with MAP speaker adaptation, the newly grown polyphone decision tree achieves superior results when used for decoding English utterances from German speakers.

# 4. Tools

## 4.1 Janus Recognition Toolkit

For all recognition experiments, the Janus Recognition Toolkit (JRTk) is used. The JRTk is developed at the Interactive Systems Laboratories at Karlsruhe Institute of Technology (KIT) and at Carnegie Mellon University in Pittsburgh [52]. Although it is originally developed for speech recognition, it has been successfully applied in many other recognition tasks such as handwriting, biosignal, emotion, and silent speech recognition. The JRTk consists of a C codebase configurable via TCL/TK scripts.

Part of the JRTk is the Janus speech recognizer. It is a HMM-based recognizer developed for research purposes. The speech recognition components such as codebooks, dictionaries, speech databases etc. are controlled via scripts and can be easily exchanged. This allows re-usage of the components which is due to the object oriented architecture of the recognizer. Details about the train procedure with Janus can be found in [53].

One of the decoders used with the JRTk is the Ibis decoder [54].

In the following work, the JRTk is used to train different accented or native AMs. The JRTk is also used for MAP or MLLR speaker adaptation of the native AM with accented speech. To decode the test and development set the Ibis decoder which is part of the JRTk is applied. To evaluate the performance of the different recognition systems, Ibis decoder takes advantage of the NIST's SCLite [55] scoring package.

## 4.2 Moses and GIZA++ SMT Tools

Moses is an open source software that allows an automatical training of SMT models [56]. It includes decoding and tuning components as well. Moses takes advantage of GIZA++ [57], which is an SMT tool used to train the IBM Models 1-5 and a word alignment model.

The training procedure with Moses starts by preparing and cleaning the bilingual corpus. To generate the bi-directional word alignments, GIZA++ is used in the

second step. The initial word alignment is then refined by using alignment heuristics such as *intersection*, *union*, *grow-diagonal*, etc. The default alignment heuristic in Moses is called *grow-diag-final*.

Having the bi-directional word alignments, the lexical translation model is computed in the next step. The phrase-table is extracted and all phrases are scored to compute the phrase translation probabilities of each phrase. All model weights and locations are then registered in a configuration file *moses.ini* which is used in the decoding process later.

For the decoding process, an LM of the target language is needed. Moses can process LMs created with the following tools: SRI LM toolkit [58], IRST [59], RandLM [60], KenLM [61]. A widely used LM format is the ARPA format supported by the SRI LM toolkit.

In this work Moses and GIZA++ SMT tools are used to translate the native words pronunciations to accented pronunciations. The SRI LM toolkit [58] is used to compute the LM probabilities for the phone n-grams. The n-gram order of the LM as well as the LM discounting strategy is defined by the user.

To tune the model weights, Moses uses the Minimum Error Rate Training (MERT) approach, described in [62].

## 4.3 Telephone-Based System for Speech Data Collection

To record accented speech in telephone quality, the Cognitive Systems Lab (CSL) telephone-based system for speech data collection was used [63]. In Figure 4.1 the architecture of the system is outlined. The system is an interactive voice response application connected to the public switched telephone network via telephone service provider. The system is connected to the speech synthesizer Festival [64], but due to quality of the produced speech, prerecorded prompts are used.

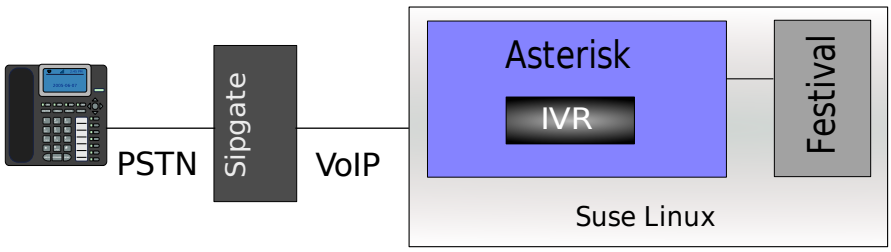


Figure 4.1: The architecture of the CSL telephone-based system for speech data collection.

The system's dialog flow is divided into two modules: One for collecting read speech and one for spontaneous speech. Due to the nature of the task, only the read speech module was used for the recordings.

# 5. Speech Corpus

The database for all experiments consists of three corpora: the English Global Phone (GP) database [1] with read speech from native American English speakers, a database with read speech from non-native speakers of English (GP-A) specially collected for the purpose of this work and the online available speech accent archive compiled at George Mason University (GMU) [2].

## 5.1 Accented Speech Data Collection

To conduct the experiments, an accented database was collected. As shown in Table 5.1 and Table 5.2, 63 non-native speakers of English (approximately 10 hours) were recorded. 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, Chinese (Mandarin or Cantonese), German and some of the languages spoken in India (Hindi, Marathi, Bengali, Telugu, Tamil). The choice of these speaker groups was based on the availability of subjects as well as on the fact that these languages are from different language families[1]. Bulgarian is from the Slavic language family, Mandarin and Cantonese are members of the Sino-Tibetan language family, German is a Germanic language and all Indian languages descend from the Indo Iranian language family [6]. Although English is an official language in India, a study described in Section 5.3 showed that the speakers from India have a specific accent and are easy to be identified as such.

The corpus consists of two parts: Read speech described in Table 5.1 and spontaneous speech described in Table 5.2.

The recorded read speech sentences are extracted from the Global Phone English database [1]. The texts are excerpts from Wall Street Journal (WSJ). The majority of topics are economy related news. All subjects were asked to read approximately 30 sentences unique for each speaker within an accent and 6 sentences that are the same for everyone. Since the inter- and intra-speaker pronunciation variability are especially emphasized in accented speech, building parallel corpora allows us to compare results and improvements.

[1] They have different scripts, use different phoneme sets, prosody, etc.

Compiling a database from non-native speakers differs in various ways from the usual approaches that are applicable for native speakers [65]. Instabilities in pronunciation of some words as well as unknown words[2] are the main obstacles the subjects stumble over. Depending on the speaker's self confidence and experience with the language, the recording of the sentences took between 30 minutes and an hour.

| | TOTAL | BG | CH | GER | IND |
|---|---|---|---|---|---|
| **# 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** | 57.4k | 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 |
| **∅ SNR*** | 11 | 12.4 | 9.6 | 11.9 | 10.1 |

Table 5.1: Statistics GP-A (read speech).

An additional task for the subjects was to speak spontaneously on a topic of their choice. For some of the subjects, it was a pleasant task and an opportunity to share some experience, story or knowledge. However, others refused to talk spontaneously for various reasons such as lack of confidence in the target language or lack of preparation or time. Example topics such as "Describe a remote control." or "Tell us about your hobby." with help questions were given to the subjects. Some of the subjects chose a different topic they feel more confident about. Therefore the topics vary from "Guitar lessons", "Movies", "School" to "Languages in India", "Area of research interests", etc.

| | TOTAL | BG | CH | GER | IND |
|---|---|---|---|---|---|
| **# speakers** | 30 | 3 | 8 | 9 | 10 |
| **male/female** | 22/8 | 1/2 | 5/3 | 8/1 | 8/2 |
| **audio length [min]** | 88 | 13 | 17 | 26 | 32 |
| **∅ time/speaker [min]** | 2.9 | 4.8 | 2.10 | 2.53 | 3.13 |

Table 5.2: Statistics GP-A (spontaneous speech).

The recording was done in a quiet room with a close-talk microphone. The sampling rate of the audio files is 16kHz in one channel. The speech was recorded and saved in waveform audio file format (WAV) with 16-bit Signed Integer PCM encoding.

[2] In read speech.

For the native speakers of Chinese and Indian languages, the speech data is also available in telephone quality. This data was recorded with the CSL telephone-based system for speech data collection [63].

Since the speech data for the Chinese and Indian databases was collected in the USA, the average time per speaker as resident in an English speaking country is much longer than for the Bulgarian and German databases. 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 databases are 4.5 months and less than a month, respectively.

All speakers are at age between 21 and 30: BG (21 - 29), CH (22 - 30), GER (22 - 30), IND(21 - 29).

An information about all languages that the subjects speak was also gathered. On average, the subjects from the Bulgarian, German and Indian databases speak 3 languages and the Chinese speakers 2 languages. Since smoking can affect the speech sounding, information about whether the subjects are smokers or not was also collected. Only 3 speakers or 4.8% of the subjects are smokers.

All recordings are made in a quiet room. The Signal-to-Noise Ratio (SNR) is calculated with the help of the CTU snr tool [66]. For this purpose, a Segmental SNR (SSNR) is used which is defined as an average of the local SNR over segments with speech activity. CTU snr tool with the argument "-est1" was used to calculate the SNR directly from the audio files, without defining segments with clean and noisy signal. SNR

The division of the speakers that is used for the experiments is as follows: 5 speakers from each accent define the test set, 5 speakers are in the development set and additional 5 speakers from 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 available also as 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.

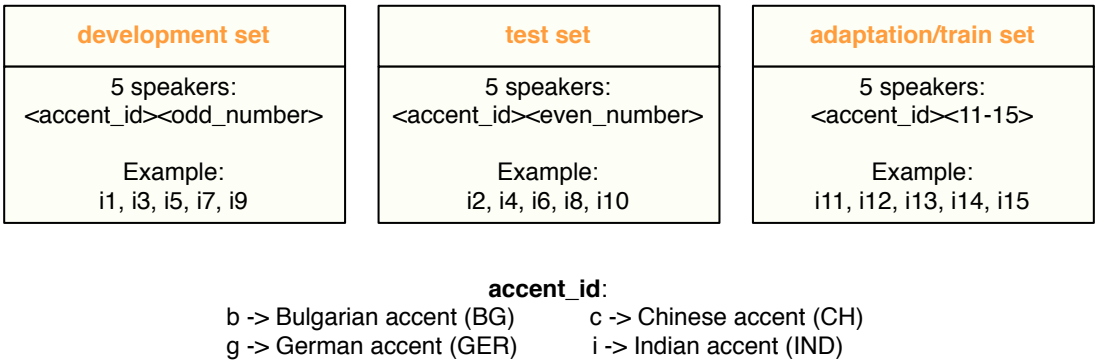


Figure 5.1: Accented corpus division.

As shown in Figure 5.1, all speakers with even numbers were used for testing (e.g. i2, i4, etc..) and all speakers with odd numbers (i1, i3, i5, etc.) formed the development

set for an accent. The 5 speakers with identifiers from 11 to 15 are used to train or adapt a system. The read texts are identical across the accents, which makes the results comparable.

## 5.2 GMU Accented Database

The GMU accented database [2] is a resource with accented speech collected at George Mason University for the purpose of studying foreign accents in English. The archive is available online and contains non-native and native accents. In this work, the phonetic transcriptions of the accented archive are used to build the SMT system that translates the native to accented pronunciations. Each speaker in the database has read the same phonetically rich paragraph:

```
Please call Stella. Ask her to bring these things with her from the
store: Six spoons of fresh snow peas, five thick slabs of blue cheese,
and maybe a snack for her brother Bob. We also need a small plastic
snake and a big toy frog for the kids. She can scoop these things into
three red bags, and we will go meet her Wednesday at the train station.
```

The read text contains 69 words, of which 55 are unique.

In October 2010, the GMU database contained approximately 1,000 speakers from approximately 300 accents from all over the world. The accents, which are from interest for us are the following Indian accents: Hindi, Marathi, Bengali, Telugu and Tamil with a total of 30 speakers in the database. From interest are also Chinese speakers of Mandarin and Cantonese (43 speakers in the GMU database), speakers from Germany (22 speakers), from Bulgaria (11 speakers) and native US English speakers, which are 115.

The GMU accented archive contains the audio files with the read sentences and their IPA transcriptions. The audio files are recorded at a sampling rate of 44.1Khz, coded with 16-bit mono. The phonetic transcriptions were produced by 2 to 4 experienced native English transcribers, following the 1996 version of the IPA chart and concentrated on the segmental transcription (no stress or tone is marked).

## 5.3 Data Analysis Using Amazon Mechanical Turk

Identifying the accent strength on phonetical level of non-native speakers is a difficult task since no standardized tests for this purpose exist. In [5], for example, formal SPEAK (Speaking Proficiency English Assessment Kit) assessment scores were used to categorize the proficiency of non-native speakers. The SPEAK test is now replaced by the TOEIC [67] (Test of English for International Communication). However, the TOEIC assesses not only the pronunciation of the foreign speech but also vocabulary, grammar, and fluency. This work is restricted to investigate acoustic changes of the spoken English due to foreign speaker origin. Making a standardized English test under supervision is not appropriate, as it also requires the willingness of the subjects to do the test.

The accent level of the speaker depends on many factors [68] such as: The age at which the speaker learned the target language, the nationality of the speaker's target

language instructor, the amount of interactive contact with native speakers of the target language and his or her learning abilities.

To categorize the level of accent of non-native English speakers as perceived by native English speakers, an online questionnaire was published. Since no unified measure of how much the phonemes of the speech are distorted by the accent, this study aimed at providing this missing information and comparing the opinion of native speakers with other measurable features that can be automatically extracted from the data.

The online questionnaire was published on the crowdsourcing platform amazon Mechanical Turk (mTurk) [69]. Crowdsourcing is the act of outsourcing tasks that have to be solved from humans to a group of volunteers. As they offer cheap and fast human labour, the crowdsourcing platforms such as mTurk are often used for research-related surveys or tasks. The tasks on mTurk are called HITs (Human Intelligence Tasks). "Turker" is the term for a worker on the crowd-sourcing platform mTurk and originates from Mechanical Turk, which is a man inside a machine pretending to be an artificial intelligence.

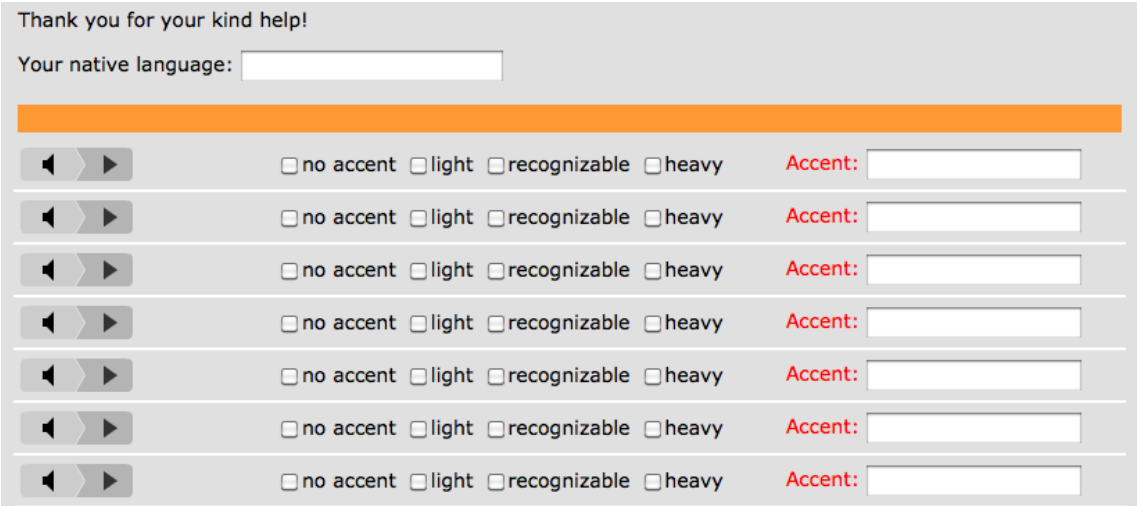


Figure 5.2: Accent Level Questionnaire

The task of the subject was to listen to the audio files. Each subject had to select the check box that best describes the accent level. As shown on Figure 5.2, there were four choices for accent levels: no accent (1), light (2), recognizable (3), and heavy (4). A text field, where the user can guess which accent was played was also included. To prevent from inferring about the current accent from the already listened audio files, the audio files were not grouped by accent.

The form was then divided in 4 HITs with 10 speakers each and was published on mTurk. To restrict the subjects to native English speakers, only people from North America were allowed to do the task. From all turkers that completed the task, only those who wrote English in the "Your native language:" text field were selected. For each accent, detailed results of the mTurk data analysis are given in Tables 5.3, 5.4, 5.5 ,and 5.6.

From 60 received assignments, 40 HITs were accepted, 7 were rejected for time reasons and 13 HITs were not included in the statistics since they were completed by non-native English speakers. The average time per HIT was 2 minutes and 58 seconds with an average hourly rate of $2.02. The questionnaire was online for approximately 20 hours.

The task of assigning a score for accent level is subjective and depends strongly on the judge's personal experience and contacts with people from the accents in question. Therefore, in crowdsourcing platforms such as mTurk it is difficult to recognize users that cheat by giving a random score for each speaker. The criterium to accept a form as valid was the time spent on solving the problem (HIT time). If it takes less than 40 seconds for a user to listen to the 10 audio samples and select the corresponding checkboxes, then the results were regarded as invalid.

To define the accent level, the following metrics were calculated: the arithmetic mean ($\mu = \frac{1}{N}\sum_{i=1}^{i=N} v_i$), the sample median ($(N/2)$th item after ordering all votes) and the most frequently selected level (peak), where $N$ is the number of votes and $v_i$ is the $i$-th vote .

| SPK | #votes | arithmetic mean | median | peak |
|---|---|---|---|---|
| b1 | 11 | 3.00 | 3 | 3 |
| b2 | 10 | 2.60 | 2 | 2 |
| b3 | 10 | 3.10 | 3 | 3 |
| b4 | 10 | 2.60 | 2 | 2 |
| b5 | 11 | 2.55 | 2 | 2 |
| b6 | 11 | 2.82 | 3 | 3 |
| b7 | 9 | 2.44 | 2 | 2 |
| b8 | 9 | 3.11 | 3 | 3 |
| b9 | 9 | 2.67 | 3 | 3 |
| b10 | 10 | 3.00 | 3 | 3 |

Table 5.3: Accent level analysis for Bulgarian speakers.

| SPK | #votes | arithmetic mean | median | peak |
|---|---|---|---|---|
| c1 | 10 | 2.70 | 3 | 3 |
| c2 | 9 | 2.89 | 3 | 2 |
| c3 | 11 | 3.27 | 3 | 4 |
| c4 | 11 | 2.18 | 2 | 2 |
| c5 | 9 | 3.11 | 3 | 3 |
| c6 | 10 | 2.80 | 3 | 3 |
| c7 | 9 | 2.67 | 3 | 3 |
| c8 | 8 | 2.75 | 2 | 2 |
| c9 | 12 | 2.67 | 3 | 3 |
| c10 | 9 | 2.78 | 3 | 3 |

Table 5.4: Accent level analysis for Chinese speakers.

| SPK | #votes | arithmetic mean | median | peak |
|---|---|---|---|---|
| g1 | 10 | 3.00 | 3 | 2 |
| g2 | 11 | 3.09 | 3 | 3 |
| g3 | 10 | 2.30 | 2 | 1 |
| g4 | 9 | 3.33 | 3 | 3 |
| g5 | 10 | 2.40 | 2 | 2 |
| g6 | 9 | 2.89 | 3 | 3 |
| g7 | 11 | 2.64 | 2 | 2 |
| g8 | 10 | 2.40 | 2 | 2 |
| g9 | 8 | 2.25 | 2 | 2 |
| g10 | 8 | 2.62 | 2 | 2 |

Table 5.5: Accent level analysis for German speakers.

| SPK | #votes | arithmetic mean | median | peak |
|---|---|---|---|---|
| i1 | 11 | 3.18 | 4 | 4 |
| i2 | 10 | 2.6 | 3 | 3 |
| i3 | 10 | 3.1 | 3 | 3 |
| i4 | 10 | 2.8 | 3 | 3 |
| i5 | 11 | 3.0 | 3 | 3 |
| i6 | 10 | 2.8 | 3 | 3 |
| i7 | 9 | 2.89 | 3 | 3 |
| i8 | 9 | 2.56 | 2 | 2 |
| i9 | 10 | 3.4 | 3 | 4 |
| i10 | 10 | 3.2 | 3 | 3 |

Table 5.6: Accent level analysis for speakers from India.

To investigate how these results correlate with the WER of every speaker, refer to Section 6.1.

As shown in Figure 5.3, the average accent level for all collected accents is approximately 2.75. This is close to the statement, that on average the people from each accent have ”recognizable” accent.

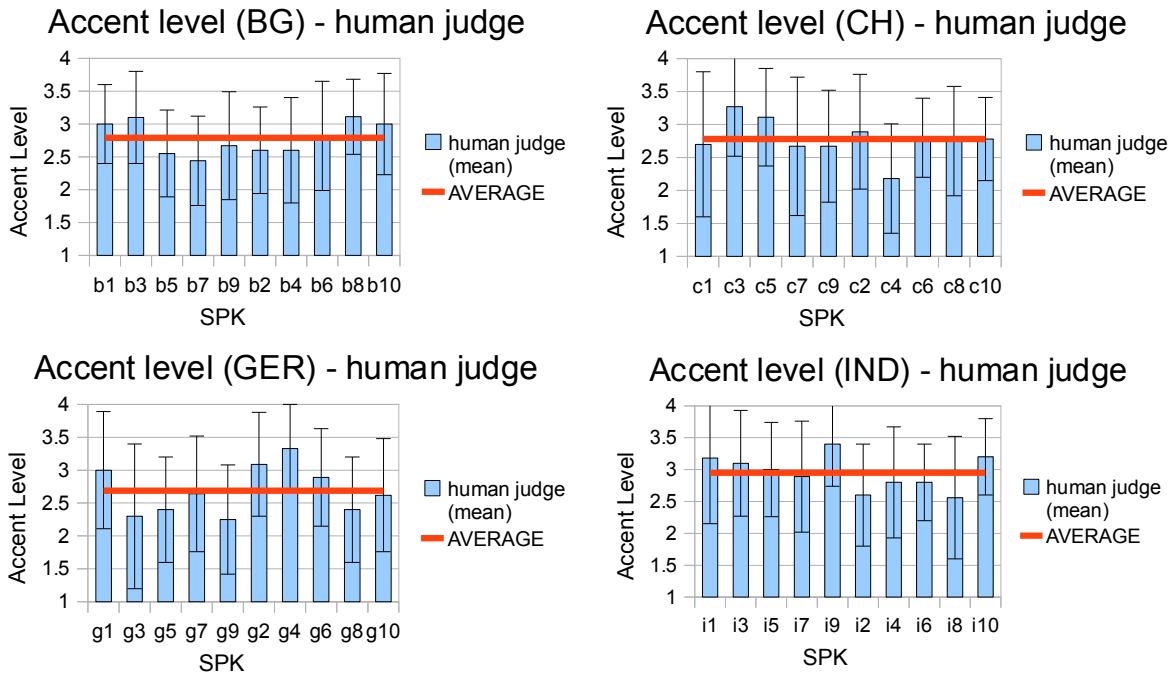


Figure 5.3: Accent level per speaker with the corresponding standard deviation value and the average accent level per accent.

According to Figure 5.3, there is a high standard deviation of the mean accent level value. This can be explained with the subjectiveness of the judgement which is not only dependent on the speakers but also on the judge's experience. The differences of the accent level between the speakers seem to be greater in the Chinese and German database than in Bulgarian and Indian database.

96% of the people who answered the question what accent they can perceive, guessed the origin of the people from India correct. The English accent originating from India may be more specific than other accents in the accented corpus. Thus it is easier to be detected. Another possible explanation may be a more frequent interaction of the turkers with people having the discussed accent.

50% of the turks recognized the accents originating from China. Only 21% detected the German accent in the accented database. The percentage was only 31% for the Bulgarian-accented English. But in this case, the turkers answered "Eastern Europe" rather than the specific country.

# 6. Experiments and Results

All experiments have the goal to reduce the WERs of the accented speech. The approaches in this work are data-driven and are therefore accent-independent. The baseline system described in Section 6.1 is trained on native American English. Two general directions to improve this baseline system to better recognize the accented English are investigated: modifying the dictionary, so that it better reflects non-native pronunciations (discussed in Section 6.2) or improving the AM to account for the new acoustic realizations that originate from the speakers' native language (Section 6.3). Section 6.3.2 describes the initial experiments on non-native speech data with telephone quality. The conclusions from all experiments are summarized in Section 6.5.

## 6.1 Baseline System

As discussed in Section 5.1, the baseline system is trained and evaluated on the GlobalPhone database - read speech from native speakers of American English. The vocabulary size is approximately 60k with about 80k dictionary entries, pronunciation variants included. The database contains recordings of 103 speakers. The total audio duration is 15.9 hours. 10 speakers from the database are used for the development set and another 10 speakers for the test set. The remaining speakers are used to train the baseline system.

Since in the experiments the accented database texts mirror the GlobalPhone texts, the same LM is used for the baseline and all accented corpora. It is a 3-gram LM with a 0.72% out-of-vocabulary (OOV) rate. The average perplexity of the LM is 31.2609.

For the acoustic modeling experiments, MFCC features are extracted from 16ms overlapping audio frames (10ms shift). They are reduced to a 42 dimensional matrix using Linear Discriminant Analysis (LDA) [70].

The emission probabilities are modeled as 2,000 fully-continuous, context-dependent quintphone models.

The context-independent baseline system has an WER of 30.51% on development and 28.43% on test set. The context-dependent system results in 13.54% WER

on development and 14.52% on test set. Due to their superior performance, only context-dependent systems are discussed further in this work.

For all recognition experiments, the Janus Recognition Toolkit (JRTk) introduced in Section 4.1 is used. The baseline system and further discussed systems are trained according to the training procedure outlined in Figure 6.1.

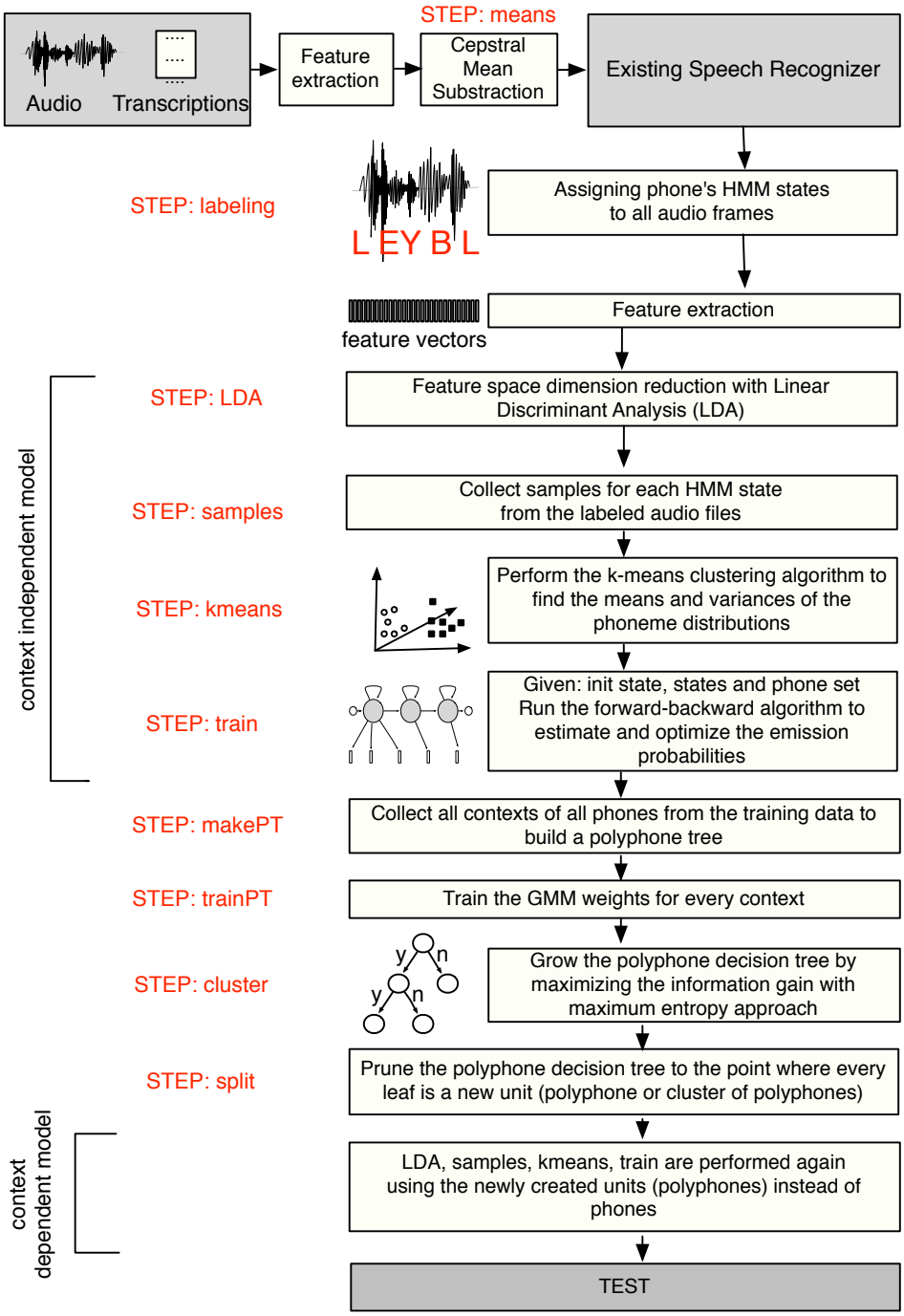


Figure 6.1: ASR training steps with JRTk.

The WERs of the baseline system confronted with non-native speech are given in Table 6.1. Although the WERs for each accent show inconsistency in the development set, they differ in the test set with only up to 1.2% absolute.

| | US ENG | BG | CH | GER | IND |
|---|---|---|---|---|---|
| **WER devSet** | 13.85% | 62.98% | 76.63% | 45.19% | 74.13% |
| **WER testSet** | 14.52% | 64.24% | 65.97% | 64.78% | 64.45% |

Table 6.1: WERs for the baseline system tested with accented speech.

To investigate the correlation between the computed WERs and the subjective accent levels assigned from the English speaking judges from the questionnaire discussed in Section 5.3, the Pearson's correlation is computed, where $cov(X, Y)$ is the covariance between $X$ and $Y$ and the $\sigma_X$ and $\sigma_Y$ are the standard deviations of those:

$$corr(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{6.1}$$

| | corr(WER,MEAN) | corr(WER,MEDIAN) | corr(WER,PEAK) |
|---|---|---|---|
| **BG** | -0.30 | -0.29 | -0.29 |
| **CH** | 0.57 | 0.81 | 0.45 |
| **GER** | 0.84 | 0.79 | 0.64 |
| **IND** | 0.22 | 0.50 | 0.37 |

Table 6.2: Correlation values between WERs and accent levels assigned by human judges.

Table 6.2 shows the correlation between the vectors with the WERs and the levels selected by the human judges. The correlation value is in the interval $[-1, 1]$. An absolute value of the correlation closer to 1 indicates stronger correlation between the vectors. The correlation of the WERs and the median of the assigned levels seems to be strong for CH, GER, IND. The correlation between the WERs and the peaks suggests a weak relationship between them. Since the median excludes the runaway values of the sample, it seems to give the best prediction for the WER from the discussed methods. However, the perception of accents is subjective and depends strongly on the judge's experience and exposure to the accent.

The anti-correlation between the WERs and the human judgement observable in Bulgarian-accented English is probably due to the fact, that this accent is not widespread. So most native American speakers may perceive the accent, but may have difficulties to judge the accent strength. Native US English speakers have greater exposure to non-native speech from the other three regions due to economic and social relationships. The Amazon Mechanical Turk questionnaire supports this assumption. Another explanation might be the low quality of the judgements which are conducted through Amazon Mechanical Turk. Due to the subjectiveness of this task, the only quality criterion for accepting the filled form as valid, is the time spent on processing the assignment (HIT).

## 6.2 Pronunciation Dictionary Experiments

The idea of adapting the pronunciation dictionary to better handle accented pronunciations is not a new one. As discussed in Chapter 3, many different research groups have tried to modify existing dictionaries using approaches they developed. The results from these approaches show minor improvements in comparison to acoustic model adaptation. In this work, the goal is to investigate if WER improvement can be achieved by modifying the existing US English dictionary using SMT models built from a small corpus of human-made IPA-based transcriptions.

For the pronunciation dictionary experiments, the following database and tools are used: Moses and GIZA++ [56] as well as the GMU accented database, described in Section 5.2.

The idea of the pronunciation dictionary experiments is outlined in Figure 6.2. The goal is to translate an existing US English dictionary into a dictionary containing accented pronunciations. For this purpose, a parallel corpus from the GMU accented database is built.

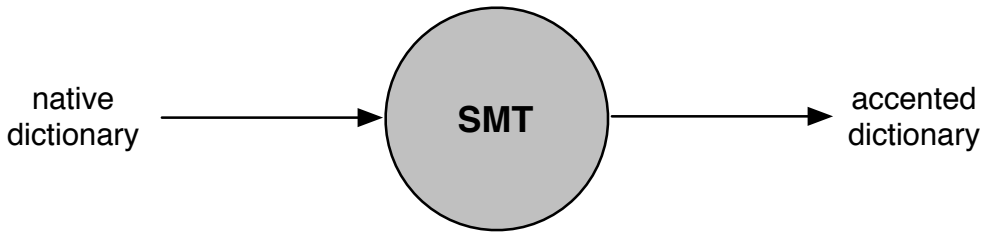


Figure 6.2: The initial idea

As discussed in Section 2.1.1, IPA contains 107 symbols and more than 50 modifiers for these. In comparison, the native English phoneme set used in the Baseline system contains 41 phonemes. Since the IPA-based transcriptions of the GMU accented database use a significantly larger phoneme set than the GlobalPhone dictionary of the Baseline system, the first step is to find a proper mapping between both phoneme sets. All 55 unique words of the GMU text:

```
"Please call Stella. Ask her to bring these things with her from the
store: Six spoons of fresh snow peas, five thick slabs of blue cheese,
and maybe a snack for her brother Bob. We also need a small plastic
snake and a big toy frog for the kids. She can scoop these things into
three red bags, and we will go meet her Wednesday at the train station."
```

are found in the GlobalPhone dictionary. Their pronunciations are used for finding the corresponding IPA-symbols according to the hand-written transcriptions for native US speakers from the GMU database. The GlobalPhone phoneme set is mapped to a simplified set of IPA symbols. For this purpose all diacritics are deleted, since they trigger exponential growth of the possible IPA phonemes. The missing IPA symbols are mapped according to [11]. In the case that many GlobalPhone phonemes correspond to one IPA symbol, the most frequently encountered correspondence is selected.

To generate a new lexicon that includes pronunciation variants of the English words as spoken by non-native speaker, Moses and GIZA++ are used. The advantage of Moses and GIZA++ is that they produce phrase-tables, which are in well-defined, human-readable format. Thus it is easy to monitor the translation process. An example of phrase table entries are:

```
...
a b ||| æ b ||| 1 0.936818 0.566667 0.15155 2.718 ||| ||| 17 30
a b ||| æ p ||| 1 0.0777041 0.1 0.00575506 2.718 ||| ||| 3 30
a b ||| æ v ||| 1 0.0108159 0.03333 0.000639451 2.718 ||| ||| 1 30
a b ||| e b ||| 1 0.105263 0.0333333 0.000798035 2.718 ||| ||| 1 30
a b ||| e p ||| 1 0.0083102 0.033333 0.000552486 2.718 ||| ||| 1 30
...
```

These rows define the model probabilities of translating ”a b” to ”æ b” . The five phrase translation scores that follow the parallel phrases are: Inverse phrase translation probability, inverse lexical weighting, direct phrase translation probability, direct lexical weighting and phrase penalty.

Two ways of building parallel corpora that might be useful for the defined task are considered: Mapping the grapheme representation of the words to the pronunciations of the GMU accented database, or using directly pronunciation-to-pronunciation mappings.

### 6.2.1 Grapheme-based Approach

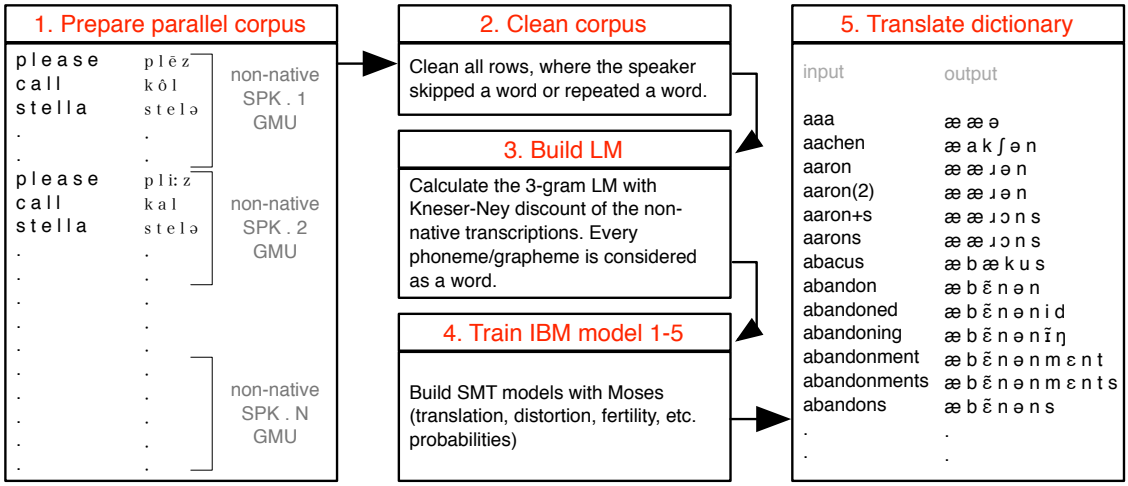


Figure 6.3: Grapheme-based SMT generation and application.

As outlined in Figure 6.3, in the grapheme-to-phoneme approach, all words from the GMU text are mapped to the IPA-based accented pronunciations of native speakers from India. The parallel corpus consists of the words' grapheme representations and

their IPA-based accented pronunciations from the GMU database. Each phoneme and grapheme is considered as a token in the SMT-based approach. After cleaning the corpus, a 3-gram LM with Kneser-Ney discounting strategy is calculated for the non-native phonemes. The parallel corpus and the LM serve as input for Moses to generate the phrase tables for the machine translation. A phrase table contains the rules and probabilities that translate a sentence from one language in another. In the case of dictionary generation, it contains a translation from the "wrong" native pronunciation to the "correct" non-native pronunciation.

All words from the initial English dictionary are translated with Moses. The weights of the SMT models in the initialization file (*moses.ini*) are set to their default values.

Generally, the grapheme-to-phoneme approach has some disadvantages: First, a grapheme-to-phoneme approach does not work well with languages like English and French, where the words are often pronounced differently from their spelling and thus the correspondence between graphemes and phonemes is weak. For example:

```
abandoned       AE B EH N AX N IY DH
abbett          AE B B IY T T
abbreviated     AE B B 9r V IH AE EH DH
abnormalities   AE B N AO M AO L IH IH IY S
abnormality     AE B N AO M AO L IH T IH
accept          AE K K IY PH T
acceptability   AE K K IY PH T AX B IH L IH T IH
acceptance      AE K K IY PH T AE N T AX
acceptances     AE K K IY PH T AE N K S
```

Common errors in the translations are phoneme deletions in the longer words. The phoneme repetitions in words containing one doubled letter are also a common error that is difficult to escape even with higher order LM[1] if the data amount is small.

Second disadvantage is that by inserting words instead of pronunciations, only one pronunciation per word is generated due to the unique grapheme representation, while many other pronunciations are possible for the same word. Third, more data is needed to cover all possible grapheme-phoneme combinations and sequences. However, if a method that selects the promising variants is used, some useful new variants can be added to the native dictionary. Filtering methods for automatically generated pronunciation variants are discussed in Section 6.2.3.

### 6.2.2 Phoneme-based Approach

In the phoneme-based approach, the goal is to capture the connection between non-accented and accented pronunciations instead of translating between grapheme representations of English words and their accented phoneme strings.

This second phoneme-based approach differs from the first approach in the initial and last steps. As outlined in Figure 6.4, instead of considering the grapheme representation of the words from the GMU text, the pronunciations from the GlobalPhone

[1] In the example a 3-gram LM is used for the translation.

dictionary are used to build the parallel corpus. In the last step, respectively, all pronunciations from the GlobalPhone dictionary are used as an input for the SMT system and are translated into their IPA-based accented version. The output is then again mapped to the GlobalPhone phonemes.

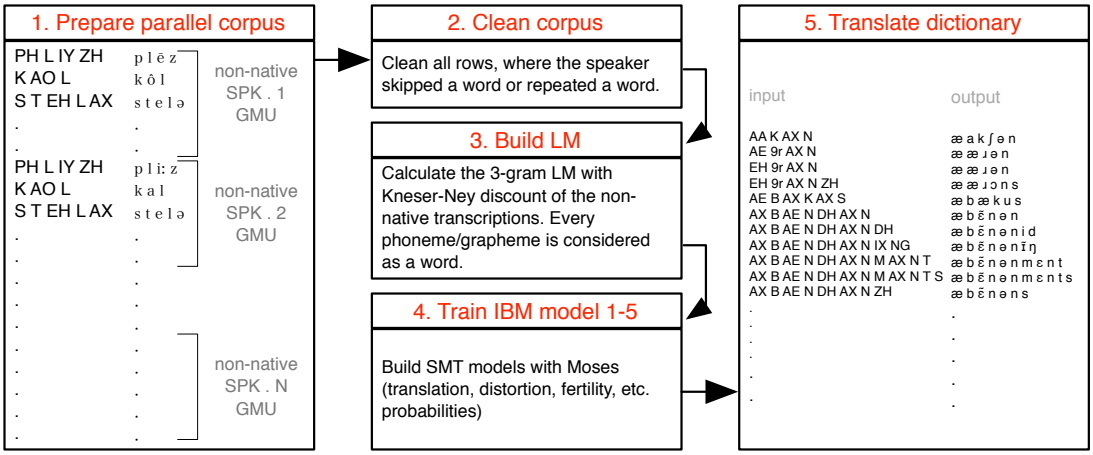


Figure 6.4: Phoneme-based SMT generation and application.

A weakness of both phoneme-based and grapheme-based approaches is the frequent switching between the phonetic alphabets. Considering that the baseline system is built from about 40 phonemes and the IPA contains more than 100 symbols, it is clear that each time a mapping between both phonetic alphabets occurs, information is lost.

To test the effectiveness of all described methods and to select the best method, the experiments are conducted with speakers from India and from China. Two dictionaries are generated using the first SMT system (SMT1): Indian-accented and Chinese-accented dictionary. To test the quality of the new dictionaries, the development sets of these accents are decoded with the native baseline system. Performance degradation for both new dictionaries is observed:

| | GlobalPhone dictionary | | SMT1 dictionary | Δ |
|---|---|---|---|---|
| **IND devset** | 74.13% | → | 78.58% | -5.6% |
| **CH devset** | 76.63% | → | 83.44% | -8.9% |

The method results in 5.6% relative performance degradation for the new Indian-accented and 8.9% for the Chinese-accented dictionary compared with the initial native US English dictionary. Since the complete dictionaries are generated from pronunciations of only 55 unique words, it is clear that they contain ”bad” variants, resulting from the lack of information about all possible phoneme combinations. For example, in the GMU text, the voiced phoneme *V* is present only in the word *five*.

In this particular case, it is often pronounced unvoiced. This results in the following translations:

```
brave            B 9r EY
obvious          AA F IY AH S
velocity         F AX L AA S AX DX IY
victorious       F IX K T AO IY AX S
victorious(2)    F IX K T OW 9r IY AX S
void             F OY DH
voter            F OW DX AX
votes            F OW T S
vouch            F AW CH
volvo            F AA L F OW
vulcan           F AH L K AX N
```

Some other words also suffer from the small data set. Unseen phoneme combinations result in variants that are difficult to pronounce. For example:

```
abrupt           AX F PH T
abruptly         AX F PH L IY
whirlpool        HH V ER L PH UW L
white            HH V IH T
white(2)         W IH DH
why              HH V AA
```

To remove the "garbage" pronunciation variants, different filtering methods described in the following section are implemented.

### 6.2.3 Filtering Methods

To develop a method that filters out the garbage pronunciation variants from all new variants, two directions are considered:

- Simple direct comparison of the original entries with the new entries.
- Using the information gained through force-aligning accented speech to pronunciations with an available speech recognizer.

The framework used to generate dictionaries containing accented pronunciations using SMT is outlined in Figure 6.5.

The first SMT system (SMT1) is built according to the phoneme-based or grapheme-based approaches explained in the previous section. The native US English dictionary is then translated with SMT1 resulting in dictionary with accented pronunciations. The accented pronunciations are filtered by using the Levenshtein distance

metric, forced-alignment information or combination of both filtering methods. The remaining new pronunciation variants are then merged with the native English dictionary to create the new dictionary, containing accented variants.

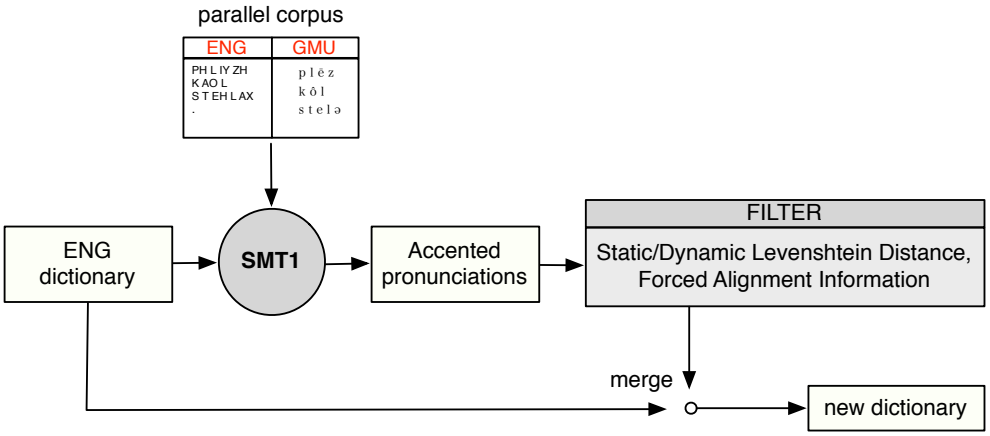


Figure 6.5: Dictionary filtering overview.

### Dynamic Distance Filtering

First, a simple filtering approach based on the Levenshtein distance (LD) as a similarity measure between new and existing variants is discussed. The idea of using LD is to compare a variant from the original dictionary with the newly created one and to delete the unchanged variants and those with a distance greater than a certain threshold.

Initially, a static upper boundary of 1 is set. Decoding with the baseline AM and this new dictionary that preserves the original US English pronunciations and contains accented variants filtered with LD boundary equal to 1, results in WER of 74.58% for the IND development set. Increasing the static LD boundary to 2 shows similar results: 74.55% WER. Compared to the original native dictionary (74.13% WER), both approaches lead to worse results.

As expected, using a static threshold for the allowed distance does not work well since shorter words are prioritized over longer ones. Therefore, mechanisms that make the boundary dependent on the word length are analyzed. All dynamically filtered dictionaries use a dynamic LD boundary (UB) which is a function of the word length. The word length (WL) is either the number of graphemes in the grapheme-based approach or the number of phonemes in the phoneme-based approach. Calculating a dynamic boundary for the allowed LD values, depending on the word length leads to improvements in the performance of the speech recognizer. An example filtering constraints are:

```
if   LD(e,n) > floor(WL/cons) or LD(e,n) == 0
     remove n from the merged dictionary
else
     keep n in the merged dictionary
```

where $e$ is the existing and $n$ is the new variant, $LD(e, n)$ is the Levenshtein distance between both variants. The LD upper boundary is a function of the word length (WL) on phoneme level and lower boundary of 0 means that the existing and the new pronunciation are identical.

In the given example, the slope of the filtering function is controlled by changing the constant (Figure 6.6). Using a $const = 3$ results in a WER of 72.82% for the IND development set given the baseline AM. This number is slightly better than the original native dictionary with WER of 74.13%.

As plotted in Figure 6.6, selecting a denominator coefficient of 3 would delete all new pronunciations with length 1 or 2, but it will accept all 12-phoneme words with a LD greater than 0 and less than or equal to 4.

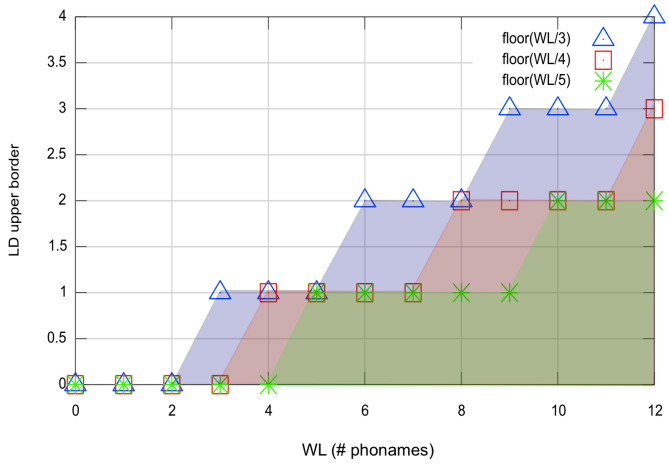


Figure 6.6: Floor functions of the WL with different denominator constants.

To analyze the new dictionaries and compare the LD-based and forced-alignment filtering, a system with ∼39 minutes transcribed data of only 5 speaker from India and different dictionaries is trained. The 5 speakers used to train the system are the same used for adaptation (IND adaptation set) and are different from the speakers in test and development set. This new system has a WER of 54.97% tested with the IND-accented development set and the GlobalPhone dictionary.

Training a new system with both native and non-native speakers results in lower WER (51.13%) due to the acoustic variety and better phoneme coverage. However, to better capture the accent-specific pronunciations, only accented data is used for training the new system.

### Forced Alignment Filtering

As mentioned in Section 6.2.3, the new pronunciation variants can be filtered by using information from the labeling step of training a speech recognizer. During labeling process, it is known which word was spoken from the transcriptions. First, the initial speech recognizer takes all pronunciations of the word that was actually spoken from the available dictionary. The task of the recognizer is to select the

most likely pronunciation among them according to the input audio. This gives information about the best fitting pronunciation of a spoken word.

One disadvantage of this approach is that the results strongly depend on the amount of selected new variants. From about 4000 (∼1000 unique) words present in the IND-accented training data, only about 200 of the new pronunciations are selected.

To increase the amount of information, an iterative approach is developed. This method is shown in Figure 6.7. It can be used if accented speech is available additionally to the accented IPA-based transcriptions. This method is first evaluated with accented English spoken by people from India.

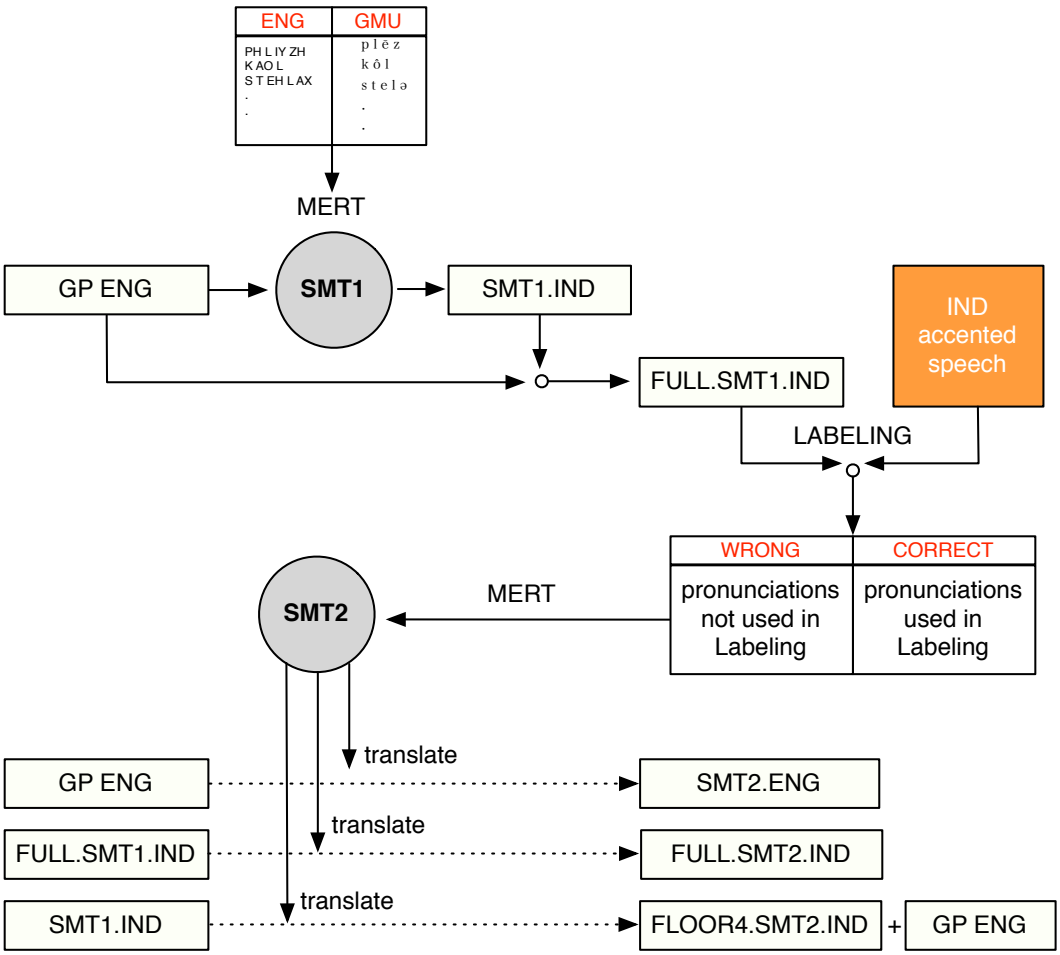


Figure 6.7: Iterative SMT approach for dictionary filtering

The idea is to build an SMT system from available phonetic transcriptions and to iteratively improve the lexicon by selecting the pronunciations that are actually chosen by a speech recognizer. As shown in Figure 6.7 the approach has the following steps:

1. A parallel corpus from the GlobalPhone pronunciations and the GMU database is created.

2. An SMT system (SMT1) is built from the parallel corpus and the SMT models' weights are adjusted by minimum error rate training (MERT) after that.

3. The SMT1 system is used to translate the original native English dictionary, which results in accented SMT1.IND dictionary.

4. Since, it is known from previous experience that the original native English dictionary contains valuable information, it is merged with the accented dictionary to create the FULL.SMT1.IND lexicon.

5. FULL.SMT1.IND is used to assign labels on accented speech from native speakers from India (IND adaptation/training set).

6. A second parallel corpus for the second SMT system (SMT2) is built as follows: The IND training set contains 4,687 words (1,637 unique). All pronunciations that are selected from the recognizer in the labeling step are considered as "correct". All alternative pronunciations of the same word, that are not used in labeling are considered to be "wrong". If a word used in the labeling process has only one pronunciation in the dictionary, this pronunciation is mapped to itself in the parallel corpus to preserve the pronunciation as it is during translation.

7. SMT2 is built from the new parallel corpus to correct the pronunciation variants from the first one.

8. The new SMT2 system is used to translate different combinations of the available dictionaries and they are tested afterwards.

The results from all filtering methods are listed in Table 6.3. Information about the dictionaries is given in Table 6.4.

| AM dictionary | Decoding dictionary | WER | Improvement |
|---|---|---|---|
| GlobalPhone ENG | GlobalPhone ENG | 54.97% | – |
| FULL.SMT1.IND | FULL.SMT1.IND | 54.12% | 1.5% |
| FULL.SMT1.IND | EXTENDED.SMT1 | 52.89% | 3.8% |
| **SMT2** | | | |
| FULL.SMT1.IND | FULL.SMT2.IND | 59.31% | -7.9% |
| FULL.SMT1.IND | SMT2.ENG | 57.18% | -4% |
| **Dynamic LD filtering** | | | |
| FULL.SMT1.IND | FLOOR3.SMT1.IND | 52.94% | 3.7% |
| FULL.SMT1.IND | FLOOR4.SMT1.IND | 52.07% | 5.3% |
| FULL.SMT1.IND | FLOOR5.SMT1.IND | 52.25% | 4.9% |
| **Combined SMT2 and Dynamic LD filtering** | | | |
| FULL.SMT1.IND | ENG+FLOOR4.SMT2.IND | 51.64% | **6.1%** |

Table 6.3: Results for different dictionary generation approaches for IND accent.

| Dictionary name | Filtering method | #initial | #new | #final |
|---|---|---|---|---|
| FULL.SMT1.IND | – | 85,308 | 55,675 | 140,983 |
| EXTENDED.SMT1 | labeling information | 85,308 | 291 | 85,599 |
| FULL.SMT2.IND | SMT2 | 124,626 | 55,599 | 124,626 |
| SMT2.ENG | SMT2 | 85,308 | 55,370 | 85,308 |
| ENG+FLOOR4.SMT2.IND | UB = floor(WL/4) | 85,308 | 23,685 | 108,993 |
| FLOOR3.SMT1.IND | UB = floor(WL/3) | 85,308 | 46,920 | 132,228 |
| FLOOR4.SMT1.IND | UB = floor(WL/4) | 85,308 | 37,924 | 123,323 |
| FLOOR5.SMT1.IND | UB = floor(WL/5) | 85,308 | 29,747 | 115,055 |

Table 6.4: Information about the generated dictionaries for IND accent.

The best improvement (6.1% relative) is achieved by using the dictionary containing the original native pronunciations and IND pronunciations after correcting with SMT2 and dynamic filtering with $UB = floor(WL/4)$. This dictionary contains 23,685 accented pronunciation variants additional to the GlobalPhone dictionary. Significantly smaller amount of new variants are added to the EXTENDED.SMT1 dictionary. They are selected by force-aligning accented audio to the available pronunciations in FULL.SMT1.IND. Dictionaries which are filtered dynamically also achieve better WERs than the native ENG dict, but at the cost of adding a significantly higher amount of pronunciation variants.

As shown in Table 6.3, FULL.SMT2.IND and SMT2.ENG perform worse. These dictionaries do not include the original ENG pronunciations as they are entirely translated with SMT2. FULL.SMT1.IND dictionary is used as input for SMT2 to generate FULL.SMT2.IND. The GlobalPhone ENG dict is translated with SMT2 to generate SMT2.ENG.

It is observed that filtering out shorter (1-3 phonemes) variants from the all new variants leads to greater improvement. This is the reason for the better performance of the floor-function compared to round-function. For example, $UB = floor(WL/4)$ removes all words with phoneme sizes of 1, 2 and 3, while the $UB = round(WL/4)$ preserves new variants with phoneme sizes of 2 or 3 and $LD = 1$.

The same experiments are conducted with the CH-accented speech data. Information about the generated CH-accented dictionaries is given in Table 6.6. The results are listed in Table 6.5. All acoustic models are trained with $\sim$ 39 minutes of speech data from the CH adaptation set and tested with the CH development set.

By adding only about 400 accented variants selected with the forced-alignment method to the initial GlobalPhone dictionary the WER gets better by 0.8% relative. However, the disadvantage of this method is that it requires accented speech data with transcriptions. The improvement is also dependent on the amount of speech data available.

| AM dictionary | Decoding dictionary | WER | Improvement |
|---|---|---|---|
| GlobalPhone ENG | GlobalPhone ENG | 55.52% | – |
| FULL.SMT1.CH | FULL.SMT1.CH | 53.89% | **2.9%** |
| FULL.SMT1.CH | EXTENDED.SMT1 | 55.09% | 0.8% |
| **SMT2** | | | |
| FULL.SMT1.CH | FULL.SMT2.CH | 61.89% | -11.5% |
| FULL.SMT1.CH | SMT2.ENG | 62.61% | -12.8% |
| **Dynamic LD filtering** | | | |
| FULL.SMT1.CH | FLOOR4.SMT1.CH | 55.04% | 0.9% |
| FULL.SMT1.CH | FLOOR5.SMT1.CH | 54.93% | 1.1% |
| **Combined SMT2 and Dynamic LD filtering** | | | |
| FULL.SMT1.CH | ENG+FLOOR5.SMT2.CH | 55.84% | -0.6% |

Table 6.5: Results for different dictionary generation approaches for CH accent.

| Dictionary name | Filtering method | #initial | #new | #final |
|---|---|---|---|---|
| FULL.SMT1.CH | – | 85,308 | 48,354 | 133,662 |
| EXTENDED.SMT1 | labeling information | 85,308 | 391 | 85,699 |
| FULL.SMT2.CH | SMT2 | 133,662 | 96,727 | 133,662 |
| SMT2.ENG | SMT2 | 85,308 | 59,163 | 85,308 |
| ENG+FLOOR5.SMT2.CH | UB = floor(WL/5) | 85,308 | 22,196 | 107,504 |
| FLOOR4.SMT1.CH | UB = floor(WL/4) | 85,308 | 34,245 | 119,553 |
| FLOOR5.SMT1.CH | UB = floor(WL/5) | 85,308 | 27,408 | 112,716 |

Table 6.6: Information about the generated dictionaries for CH accent.

Small performance improvement for the CH speech data is gained with the simple filtering methods. This is, however, at the cost of adding many new variants to the dictionary which increases the decoding time.

The iterative SMT approach does not work for the Chinese accent as it works for the Indian accent. Since for both accents different methods show different improvement patterns, the SMT-based approach is not accent independent.

## 6.3 Acoustic Modeling Experiments

In this section acoustic adaptation experiments are discussed. Combined acoustic adaptation with dictionary modification is discussed in Section 6.4.

In Section 6.3.1 the acoustic adaptation experiments with close-talk microphone data are described. Initial experiments for the Indian and Chinese accents with telephone-quality data are discussed in Section 6.3.2.

### 6.3.1 Close-Talk Microphone Speech Data Experiments

As concluded in [5], [30] and [47], a great improvement of the speech recognition performance can be achieved with speaker adaptation. Therefore, the baseline speech

recognizer is adapted, using the two techniques discussed in Section 2.1.7. The results from testing with the development set are shown in Table 6.7.

| | MAP (with 5 speakers) | MLLR (with 5 speakers) |
|---|---|---|
| **WER BG devSet** | 51.59% | 52.07% |
| **WER CH devSet** | 55.56% | 58.94% |
| **WER GER devSet** | 37.49% | 40.02% |
| **WER IND devSet** | 45.88% | 57.37% |

Table 6.7: WERs for the accented speech after MAP adaptation and MLLR adaptation of the baseline system.

The Indian accented data used for adaptation is approximately 39 minutes of speech from 5 Indian speakers. The adaptation data from native Bulgarian speakers is approximately 40 minutes, from native German speakers approximately 38 minutes and from native Chinese speakers approximately 46 minutes. The data for all accents contains the same utterances read from the same number of speakers.

The experiments show that in all cases except for the Bulgarian development, applying MAP adaptation after MLLR adaptation gives superior improvements in speech recognition performance. A possible reason for this exception might be the inconsistencies of the accent. As shown in Table 6.7, the WER after MAP adaptation is lower than the WER of MLLR adaptation for all accents. But for Bulgarian accent, the difference of 0.48% absolute is very small compared to the 11.49% absolute difference on the IND development set.

The results of combining the two adaptation techniques are listed in Table 6.8. For each accent (except for Bulgarian) applying MAP over MLLR improves the system accuracy, while applying MLLR after MAP adaptation shows slight degradation compared to applying only MAP adaptation.

| | MAP over MLLR (5 speakers) | MLLR over MAP (5 speakers) |
|---|---|---|
| **BG devSet** | 54.44% | 51.85% |
| **CH devSet** | 49.03% | 56.51% |
| **GER devSet** | 35.12% | 37.70% |
| **IND devSet** | 38.74% | 46.42% |

Table 6.8: WERs for applying two adaptations (MAP and MLLR) in sequence.

To investigate, if further improvement is gained with more adaptation data, the test set for each accent is added to the adaptation set. MAP adaptation is then applied over MLLR. The results from testing on the development sets are shown in Table 6.9.

| | MAPoMLLR (10) | MAPoMLLR (5) | Δ(0-5) | Δ(5-10) |
|---|---|---|---|---|
| **BG devSet** | 43.62% | 54.44% | 13.6% | 19.9% |
| **CH devSet** | 42.02% | 49.03% | 36.0% | 14.3% |
| **GER devSet** | 29.63% | 35.12% | 22.3% | 15.6% |
| **IND devSet** | 34.00% | 38.74% | 47.7% | 12.2% |

Table 6.9: WERs of adaptation with 10, 5 speakers and the relative gains.

As the results of speaker adaptation with more accented data (10 spekers) indicate, Bulgarian and German accents gain most improvement the next 5 speakers for adaptation. After the initial gain of 47.7% relative from adaptation with 5 speakers, the next 5 speakers give an improvement of only 12.2% on the IND development set. The results from Table 6.9 show that the accents that initially gain the biggest improvement from adaptation with 5 speakers, get fewer improvement when adapted with more data. This might be a result of the "consistency" of an accent: Since the IND accent is very specific and can be detected very quickly from a native English speaker (as discussed in Section 5.3), a small amount of IND-accented adaptation data already leads to major improvements. For accents where the acoustic errors are not consistent, the gains from speaker adaptation come with greater amount of speech data, whereas accents with consistent errors reach a point of saturation after small amounts of data.

To analyze if there is a tendency in WER improvement when adapted with different numbers of speakers, the development set of each accent is tested additionally with systems adapted with 2, 4, 6 and 8 speakers. The outcomes of this experiment is shown in Figure 6.8. The results for Chinese and Indian accents look very similar with a very steep slope at the beginning and relatively smooth improvement after that. For Bulgarian accent slight improvement is visible with less data while the improvement jump comes between the 4th and 6th speaker, reducing the WER consistently after that. The curve for the German accent shows fluctuations from the beginning to the end but the general tendency is towards a lower error rate.

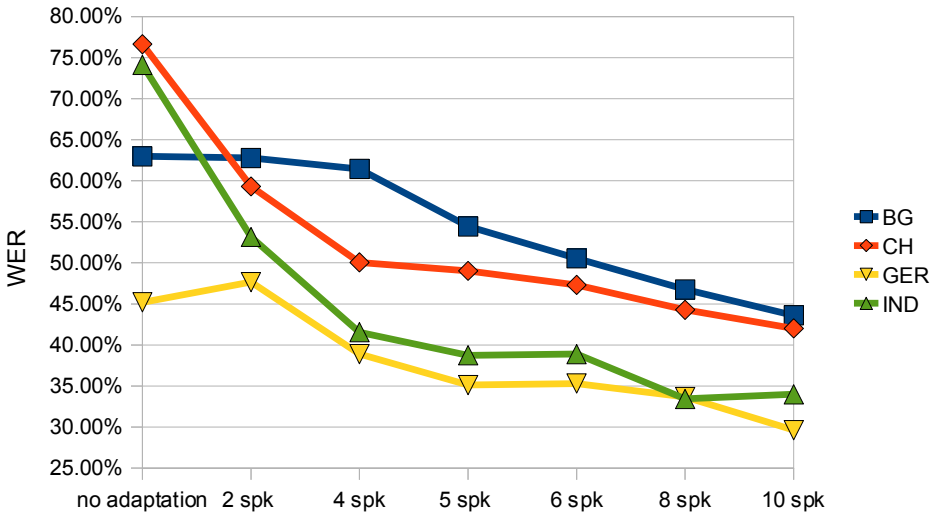


Figure 6.8: WER improvement tendency with increasing number of speakers.

The improvement curves plotted in Figure 6.8 might behave differently with another data ordering. To check if the characteristics of the curves depend on the order of adaptation speakers, the adaptation sets are reversed as follows: A system is adapted with the last 5 adaptation speakers from each accent. For example, if the results in Figure 6.8 are achieved with adapting the baseline with speaker 1, 2, 3, .. 10 in this order, now an adaptation with speaker 6 to 10 is done first. Figure 6.9 shows that the improvement tendency remains, even if 5 different speakers are first used for adaptation.

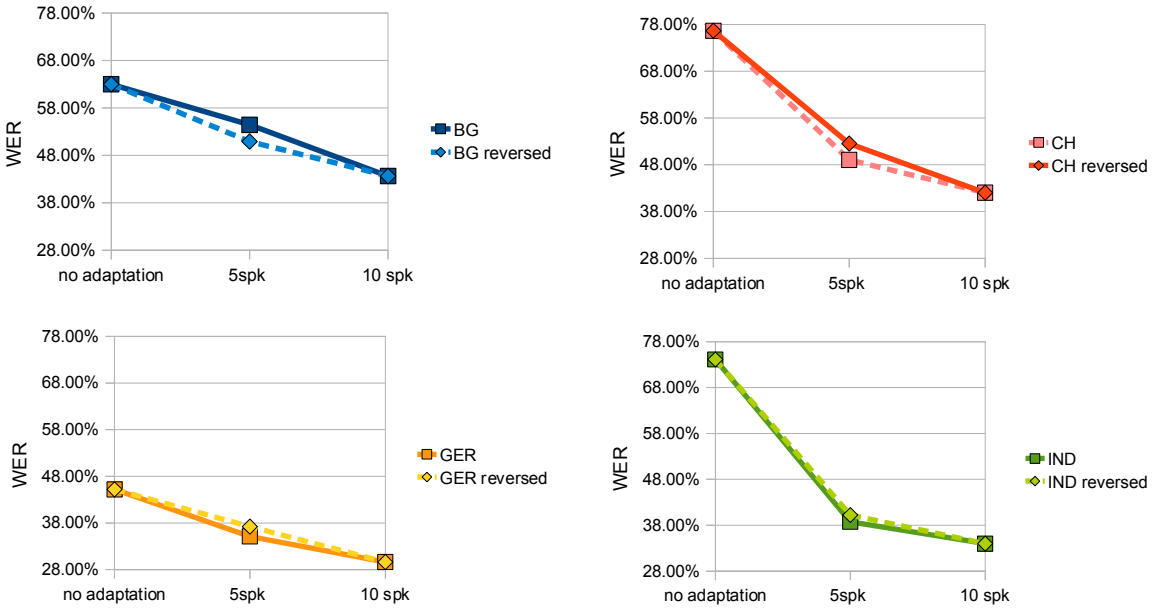


Figure 6.9: WER improvement by adapting each accent with 5 different speakers.

In Figure 6.10, the overall WERs from the different adaptation methods are compared. The bigger the WER of the baseline system, the bigger gains are observed after adaptation. The consistent pronunciations of the speakers from India seems to help for better results after speaker adaptation and this accent gains the highest improvement from acoustic adaptation.

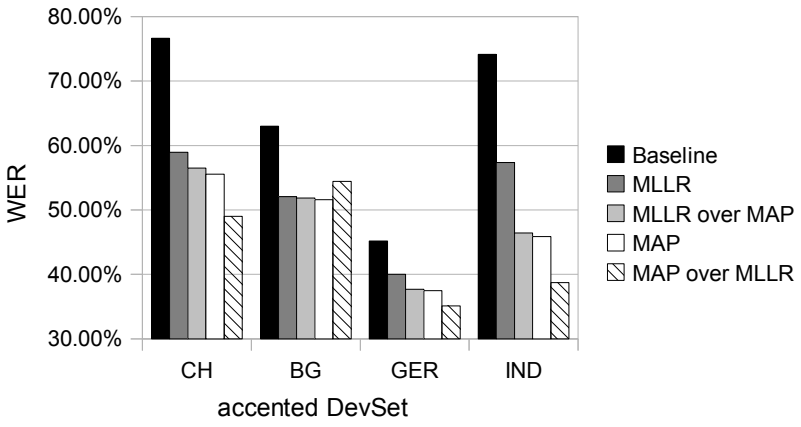


Figure 6.10: Overall improvements from acoustic adaptation approaches on development sets (5 speakers).

### 6.3.2 Initial Telephone Speech Data Experiments

In this section, the initial results from the experiments with the available accented speech data in telephone quality are described.

Since the native English speech is not available in telephone quality, the audio files used to build the baseline system from Section 6.1 are resampled to 8kHz. As seen in Table 6.10, the new baseline system has a performance of 17.14% on the test set, which is a 15.3% relative degradation compared to the higher quality baseline. This difference is partly due to the downsampling from 16kHz to 8kHz. The feature extraction window and shift (discussed in Section 2.1.2) remain unchanged to the ones used in the 16kHz baseline. This is another reason for the higher performance degradation.

| | close-talk microphone | telephone quality |
|---|---|---|
| **Baseline TestSet** | 14.52% | 17.14% |
| **Baseline DevSet** | 13.85% | 19.66% |
| **IND TestSet** | 64.45% | 77.59% |
| **IND DevSet** | 74.13% | 81.43% |
| **CH TestSet** | 65.97% | 68.95% |
| **CH DevSet** | 76.63% | 79.49% |

Table 6.10: IND- and CH-accented data in close-talk microphone and telephone quality.

After adapting the resampled baseline system with accented telephone speech a significant improvement is observed. The results for the Indian and Chinese accents are given in Table 6.11.

| | telephone quality | MAP over MLLR (telephone quality) |
|---|---|---|
| **IND TestSet** | 77.59% | 44.84% |
| **IND DevSet** | 81.43% | 58.78% |
| **CH TestSet** | 68.95% | 61.14% |
| **CH DevSet** | 79.49% | 56.86% |

Table 6.11: Results of IND- and CH-accented data with telephone quality after acoustic adaptation.

## 6.4 Combining Acoustic and Lexical Modeling

The biggest WER reduction is expected to be a result of a combined acoustic and lexical accent adaptation. For this purpose, the most successful methods of the acoustic and pronunciation dictionary adaptations are combined. By generating new pronunciation variants with SMT1, filtering the unsuccessful ones with dynamic upper bound for the Levenshtein distance and adding them to the initial US English dictionary, a small improvement of the recognition accuracy is observed for all accents. The reason to select simple filtering based on Levenshtein distance is

the consistent improvement of this method observed in the lexical experiments with Indian and Chinese accented data. The sizes of the new dictionaries are listed in Table 6.12. The WERs by adapting and then decoding with these new lexicons are given in Table 6.13. The best scores are highlighted in red.

| Accented dictionary | Number new variants added |
|---|---|
| SMT1.FLOOR4.BG | 33,148 |
| SMT1.FLOOR4.CH | 33,537 |
| SMT1.FLOOR4.GER | 33,281 |
| SMT1.FLOOR4.IND | 31,730 |

Table 6.12: New pronunciation variants for each accent after translating with SMT1 and filtering with dynamic LD FLOOR.4 function.

| | MAPo.MLLR | MAPo.MLLR+D | MAP | MAP+D | MLLR | MLLR+D |
|---|---|---|---|---|---|---|
| **BG** | 54.44% | 54.38% | 51.59% | **51.45%** | 52.07% | 51.43% |
| **CH** | 49.03% | **48.36%** | 55.56% | 55.24% | 58.94% | 58.62% |
| **GER** | 35.12% | **34.32%** | 37.49% | 38.00% | 40.02% | 39.06% |
| **IND** | 38.74% | **38.08%** | 45.88% | 46.36% | 57.37% | 56.43% |

Table 6.13: WERs after applying adaptation with the new dictionaries.

As given in Table 6.13, for two cases (MAP adapted German and Indian systems) the new dictionaries do not outperform the initial GlobalPhone dictionary. In all other cases, the systems with the new dictionaries perform slightly better. This might be a consequence of adding many new variants.

However, the positive effect of dictionary modification is weakened if transferred to the adapted native baseline system. This is logical since both approaches are not completely complementary. With acoustic adaptation the phoneme models are shifted towards what they sound as spoken from non-native speaker. However, the acoustic adaptation covers only the phoneme substitution cases. A promising way of combining lexical and acoustic adaptation is to keep in the dictionary only the new variants that contain at least one insertion or deletion. This reduces the amount of new pronunciation variants. Therefore, two sets of dictionaries are compared:

(1) GlobalPhone ENG + accented pronunciation filtered with $floor(word\_length/4)$.

(2) Same as (1), but containing only variants with at least one insertion or deletion.

In Table 6.14 both methods are compared. All dictionaries from set (2) contain between 88,000 and 93,000 entries (3000-8000 new variants).

| | Baseline | A | A + method (1) | A + method (2) |
|---|---|---|---|---|
| **BG** | 62.98% | 54.44% | 54.38% | **54.14%** |
| **CH** | 76.63% | 49.03% | 48.36% | **48.00%** |
| **GER** | 45.19% | 35.12% | **34.32%** | 35.12% |
| **IND** | 74.13% | 38.74% | **38.08%** | 38.56% |

Table 6.14: Results for decoding the development sets with adapted Baseline (A) and SMT1.FLOOR4 dictionaries with and without variants containing only substitution.

For German and Indian accents, the dictionaries created with method (1) perform best evaluated with the development set. For the Bulgarian and Chinese accent, method (2) shows better results. Therefore, the dictionaries from set (1) are used in the evaluation of the German and Indian test sets and from set (2) for the Bulgarian and Chinese test sets.

| | Baseline | Lexical adaptation | Acoustic adaptation | Combined adaptation |
|---|---|---|---|---|
| **BG** | 64.24% | 64.79% | 47.33% | **46.95%** |
| **CH** | 65.97% | 65.34% | 44.79% | **44.10%** |
| **GER** | 64.78% | 62.83% | 44.89% | **44.74%** |
| **IND** | 64.45% | 62.61% | 30.77% | **30.19%** |

Table 6.15: Results from decoding the test sets.

As shown in Figure 6.11, the system adapted for Indian accent performs best. Similar to what is observed from the experiments with the development sets, Indian accent gained the biggest improvement.

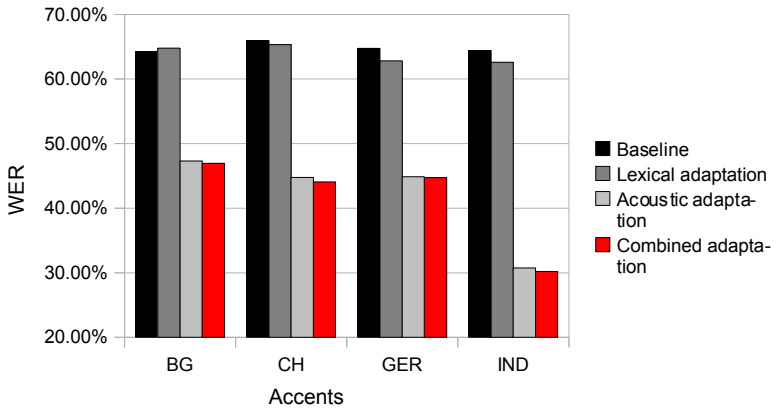


Figure 6.11: Performance of the ASR systems on the test set.

The best results are achieved by combining lexical and acoustic adaptation. However, the highest WER reduction comes from the acoustic adaptation: 23-55% relative gain.

In Figure 6.12 the effects of acoustic and combined adaptation per speaker are plotted.

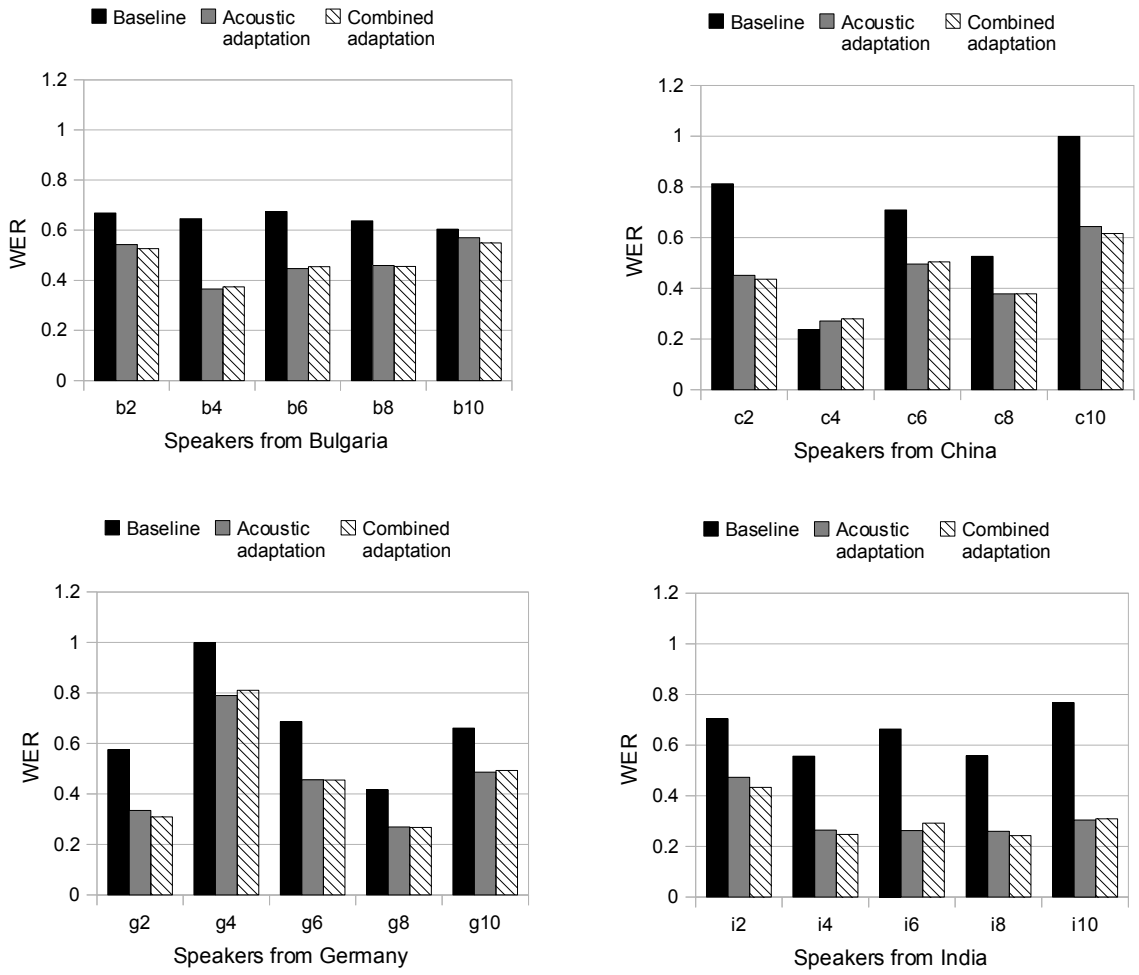


Figure 6.12: Performance improvement for each speaker.

The Bulgarian speakers have very similar baseline WERs. However, they show very different improvement rates when the baseline system is adapted.

As visualized in Figure 6.12, the baseline performance confronted with non-native English from Chinese speakers shows a high variability. Almost all speakers gain from the acoustic and combined adaptations, except speaker $c4$. The initial WER for this speaker is very low, which means, that he or she does not have a strong accent. With 22%, the WER of $c4$ is close to those of the native US English speakers and therefore accent adaptation is harmful in this case.

The initial WERs of the German speakers show also a high variability. However, even the speakers with initially low WER show a high improvement after acoustic adaptation. Dictionary modification, however, results in WER reduction only for one of the speakers. For two of the speakers the performance decreases after lexical adaptation.

All speakers from India gain significant improvement from acoustic adaptation. The new dictionary increases the recognition accuracy for 3 of the 5 speakers.

## 6.5 Summary

The initial concept of the lexical modeling experiments in this work was to translate the pronunciations of an existing US English dictionary into accented pronunciations. For this purpose an SMT system was built from human-made, IPA-based transcriptions of accented speech.

In the process of implementing this idea, the following problems were faced:

- Small amount of accented speech data and transcriptions.
- Many garbage pronunciations produced from the SMT system.
- Different phonetic alphabets.

Some interesting questions emerged:

- Is the grapheme-to-phoneme or phoneme-to-phoneme approach better for the task?
- How to remove the pronunciations that contain many translation errors from the new dictionaries?
- How to employ a speech recognizer to better select fitting from non-fitting pronunciations?

The experiments showed that phoneme-to-phoneme translations work better than grapheme-to-phoneme translations for English. The dictionaries automatically generated with SMT methods from a small amount of accented transcriptions were filtered to gain improvement. Otherwise they resulted in performance degradation. A simple edit distance filtering resulted in up to 6% relative WER improvement. Employing an existing speech recognizer to select the successful pronunciation variants lead to improvements with a smaller amount of new pronunciation variants.

For the acoustic adaptation, the following points were of interest:

- How does the amount of accented acoustic data relate to WER improvement?
- Are some accents with more consistent pronunciations than other accents?

The more adaptation data is available, the bigger improvement was seen. However, the accents that gained the biggest improvement with the first 5 speakers, got smaller relative improvements with the next 5 speakers for adaptation. This might be related to the consistency of an accent. In accents where the pronunciation of words is similar for each speaker, smaller amounts of accented data lead to high improvement.

Finally, to gain the greatest improvement, the lexical and acoustic approaches were combined. To achieve most benefits is a challenge since both methods are not complementary.

# 7. Conclusion and Future Work

Accented speech is a challenge for the existing ASR technologies. One of the major problems is that accented pronunciations vary across speakers and speaker groups. Additional difficulty is the speaker's lack of confidence when pronouncing a foreign word, which results in varying pronunciations produced by the same speaker. The proposed thesis investigates lexical and acoustic modeling techniques that influence the speech recognition performance of accented English.

## 7.1 Results

For this work, a corpus of accented English from four different language groups was collected: Native speakers of Chinese, Bulgarian, German and Indian languages. The corpus consists of approximately 10 hours of read and spontaneous speech from 63 non-native English speakers. The accented utterances are also available from native US English speakers in the GlobalPhone English database [1].

The accent level of the speakers from the collected corpus was analyzed using a questionnaire published on Amazon Mechanical Turk, targeting native US English speakers. According to the participants in the survey, on average the speakers have a recognizable accent. The corpus was used to train, adapt and test ASR systems with the goal to improve the speech recognition performance for non-native English.

It was observed that the standard techniques used for recognition of accented speech MAP and MLLR perform well even with small amounts of data. Using more accented speech for adaptation further reduces WERs. The reason is that the phonemes are pronounced differently from non-native speakers which can be handled with acoustic adaptation techniques. WERs are reduced by 25-50% relative compared to an ASR system trained on purely native speech.

A new approach for generating accented pronunciations was presented in this work. The approach uses SMT systems built from the phonetic IPA-based transcriptions in GMU database [2] to generate non-native pronunciations. Up to 6% relative WER reduction was achieved by modifying the existing US English dictionary. Due to the restricted word set and the small amount of accented phonetic transcriptions

for the target accents, many ”garbage” pronunciations were produced. To filter the ”garbage” pronunciations of the SMT output, different methods were investigated. Their advantages and disadvantages are:

- Simple filtering based on Levensthein distance:
  - ⊕ Gives small improvements consistently when filtering function and its parameters are carefully selected.
  - ⊕ Does not require accented speech data.
  - ⊖ Many new wrong variants added.
- Using the pronunciation variants selected with ASR system
  - ⊕ Higher improvement with smaller amount of new variants compared to the other approach.
  - ⊖ Requires accented audio data.
  - ⊖ Results strongly depend on amount of selected new variants occurring in the test set, respectively on the amount of available speech data.

Possible reasons for the small WER improvement with the dictionary experiments are the small data set and the information loss due to the conversion between the phonetic alphabets. Given that the human-made accented transcriptions are perfect, a set of 55 words can hardly cover all possible or distinctive phoneme combinations for the accented pronunciations. Simplifying the IPA symbols used for the transcription by removing the diacritics and sound length, tone, intonation symbols to better cover the initial phoneme set also results in loss of features that might be important for the accent.

Combining the acoustic and lexical modeling approaches resulted in the systems with the best WERs. However, the improvement from the lexical adaptation was very small. Overall relative WER reduction on evaluation sets from the combined lexical and acoustic adaptation is as follows: 26.9% for Bulgarian, 33.2% for Chinese, 30.9% for German and 53.2% for Indian accents.

It was observed that Indian and Chinese accents gain improvement faster with acoustic adaptation. Indian accented English speech was easily identified by native US English speakers from the Amazon Mechanical Turk accent level questionnaire. This may suggest that the speakers of Indian languages pronounce the words consistently and therefore smaller amount of data leads to higher improvements than observed with the other accents. The confident English pronunciations of the speakers from India might result from the fact, that English is an official language in India and children are exposed to the specific accent early in their lives. Indian accent gained the biggest improvement compared to the other accents in both acoustic and lexical adaptation. The Chinese accent was recognized by 50% of the native English speakers that made a guess. This accent gained higher improvement from the adaptations compared to German and Bulgarian accents.

## 7.2 Future Research Directions

In future work, experiments with different metrics and functions that better filter the garbage pronunciations can be conducted. In the context of lexical adaptation, rules extracted automatically from the phonetic transcriptions can be applied and evaluated. An interesting question is, if lexical adaptation can gain improvements in the case when acoustic adaptation does not further improve the speech recognition performance?

In context of acoustic adaptation, the PDTS technique [50] can be used to complement the mandatory speaker adaptation.

Further investigation of the accent consistency with more data and more speakers can be made. Interesting questions are: Do some accents gain faster improvement from the adaptation techniques than others and why? At what point does more adaptation data no longer improve WER? How much data is needed to achieve the performance of speech recognizers on native speech?

An interesting question is if a further improvement can be gained by using additional semantic and syntactic information on the top of acoustic and language modeling. This information is crucial for human beings to compensate for imperfect acoustic information, therefore highly relevant in the accented speech. Experiments with the GP-A spontaneous speech database can answer the question if grammar constructs from the native language are transferred to the non-native spontaneous speech.

According to [71] some accents have strong acoustic level relationship. Examining the similarities and differences between accents can help in overcoming the problem of data sparseness in the context of non-native speech.

# Bibliography


[1] T. Schultz, GlobalPhone: a Multilingual Speech and Text Database Developed at Karlsruhe University, in *ICSLP*, pp. 345–348, 2002.

[2] S. Weinberger, The Speech Accent Archive. accent.gmu.edu . George Mason University, 2010.

[3] L. M. Tomokiyo, Handling Non-native Speech in LVCSR: A Preliminary Study, in *InSTIL*, 2000.

[4] B. T. Conboy and P. K. Kuhl, Impact of second-language experience in infancy: brain measures of first- and second-language speech perception, in *Developmental Science*, 2010.

[5] L. M. Tomokiyo and A. Waibel, Adaptation Methods for Non-Native Speech, in *Multilinguality in Spoken Language Processing*, 2001.

[6] C. F. Meyer, *Introducing English Linguistics*, Cambridge University Press, 2009.

[7] D. Crystal, *Andreas Gardt and Bernd Hüppauf (eds), Globalization and the future of German (Berlin: Mouton de Gruyter)* , 27 (2004).

[8] S. Young, Large Vocabulary Continuous Speech Recognition: a Review, in *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 3–28, 1996.

[9] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, Prentice Hall, 2000.

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

[11] Arpabet: http://en.wikipedia.org/wiki/Arpabet.

[12] cmudict: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[13] H. Hermansky, Perceptual linear predictive (PLP) analysis of speech, in *ASA*, volume 87, pp. 1738–1752, 1990.

[14] L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, in *IEEE*, pp. 257–286, 1989.

[15] S. Young, J. Odell, and P. Woodland, Tree-Based State Tying for High Accuracy Acoustic Modelling, in *HLT*, pp. 307–312, 1994.

[16] S. F. Chen and J. Goodman, An Empirical Study of Smoothing Techniques for Language Modeling, in *Computational Linguistics*, pp. 310–318, 1998.

[17] D. B. Paul, An Efficient A* Stack Decoder Algorithm for Continuous Speech Recognition with a Stochastic Language Model, in *Workshop on Speech and Natural Language*, Lincoln Laboratory, MIT, 1991.

[18] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing: a guide to theory, algorithm, and system development*, Prentice Hall, 2001.

[19] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education, publ. as Prentice Hall, 2003.

[20] J. Bilmes and K. Kirchhoff, Factored Language Models and Generalized Parallel Backoff, in *HLT*, 2003.

[21] M. J. F. Gales, Maximum Likelihood Linear Transformations for HMM-based Speech Recognition, in *Computer Speech and Language*, volume 12, pp. 75–98, 1998.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubimn, Maximum Likelihood from Incomplete Data via the EM Algorithm, in *Journal of the Royal Statistical Society*, volume 39, pp. 1–38, 1976.

[23] J.-L. Gauvain and C.-H. Lee, MAP Estimation of Continuous Density HMM: Theory and Applications, in *In: Proceedings of DARPA Speech and Natural Language Workshop*, pp. 185–190, 1992.

[24] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, A Statistical Approach to Machine Translation, in *Computational Linguistics*, volume 16, pp. 79–85, 1990.

[25] K. Knight, A Statistical MT Tutorial Workbook, 1999.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, in *Computational Linguistics*, pp. 311–318, 2002.

[27] G. Doddington, Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics, in *HLT*, pp. 138–145, 2002.

[28] S. Banerjee and A. Lavie, METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments, in *ACL Workshop*, pp. 65–72, 2005.

[29] F. J. Och, Minimum Error Rate Training in Statistical Machine Translation, in *41st Annual Meeting of the Association for Computational Linguistics*, pp. 160–167, 2003.

[30] L. M. Tomokiyo, *Recognizing non-native speech: Characterizing and adapting to non-native usage in LVCSR*, PhD thesis, School of Computer Science, Language Technologies Institute, CMU, 2001.

[31] TIMIT: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1.

[32] IViE corpus: http://www.phon.ox.ac.uk/files/apps/IViE/.

[33] M. Raab, R. Gruhn, and E. Nöth, Non-Native Speech Databases, in *ASRU*, pp. 413–418, Kyoto (Japan), 2007.

[34] CSLU Foreign Accented Corpus: http://cslu.cse.ogi.edu/corpora/fae/.

[35] W. Menzel, E. Atwell, P. Bonaventura, D. Herron, P. Howarth, R. Morton, and C. Souter, The ISLE corpus of non-native spoken English, in *LREC*, pp. 957–963, Athens, Greece, 2000.

[36] R. Gruhn, T. Cincarek, and S. Nakamura, A multi-accent non-native English database, in *ASJ*, pp. 195–196, 2004.

[37] M. Lincoln, I. McCowan, J. Vepa, and H. K. Maganti, The Multi-Channel Wall Street Journal Audio Visual Corpus (MC-WSJ-AV): Specification and Initial Experiments, in *ASRU*, pp. 357–362, 2005.

[38] K. J. Engen, M. Baese-Berk, R. E. Baker, A. Choi, M. Kim, and A. R. Bradlow, The Wildcat Corpus of Native- and Foreign-accented English: Communicative Efficiency across Conversational Dyads with Varying Language Alignment Profiles, in *Language and Speech*, 2010.

[39] AMI corpus: http://corpus.amiproject.org/.

[40] W. Byrne, E. Knodt, S. Khudanpur, and J. Bernstein, Is Automatic Speech Recognition Ready for Non-Native Speech? A Data Collection Effort and Initial Experiments in Modelling Conversational Hispanic English, in *ESCA Conference on Speech Technology in Language Learning*, pp. 37–40, 1998.

[41] L. M. Tomokiyo, Lexical and Acoustic Modeling of Non-native Speech in LVCSR, in *ICSLP*, pp. 346–349, 2000.

[42] S. Goronzy and K. Eisele, Automatic Pronunciation Modeling for Multiple Non-Native Accents, in *ASRU*, pp. 123–128, 2003.

[43] J. J. Humphries, P. Woodland, and D. Pearce, Using Accent-Specific Pronunciation Modelling for Robust Speech Recognition, in *ICSLP*, pp. 2324–2327, 1996.

[44] J. J. Humphries and P. C. Woodland, The Use of Accent Specific Pronunciation Dictionaries in Acoustic Model Training, in *ICASSP*, pp. 317 – 320, 1998.

[45] I. Amdal, F. Korkmazskiy, and A. C. Surendran, Data-Driven Pronunciation Modelling for Non-Native Speakers using association strength between phones, in *ASRU*, 2000.

[46] D. Vergyri, L. Lamel, and J.-L. Gauvain, Automatic Speech Recognition of Multiple Accented English Data, in *INTERSPEECH*, 2010.

[47] G. Bouselmi, D. Fohr, I. Illina, and J. P. Haton, Fully Automated Non-Native Speech Recognition Using Confusion-Based Acoustic Model Integration and Graphemic Constraints, in *ICASSP*, 2006.

[48] M. Raab, R. Gruhn, and E. Nöth, Multilingual Weighted Codebooks for Non-Native Speech Recognition, in *Proc. TSD*, pp. 485–492, 2008.

[49] Z. Wang, T. Schultz, and A. Waibel, Comparison of Acoustic Model Adaptation Techniques on Non-Native Speech, in *ICASSP*, pp. 540–543, 2003.

[50] T. Schultz and A. Waibel, Polyphone Decision Tree Specialization for Language Adaptation, in *ICASSP*, volume 3, pp. 1707–1710, 2000.

[51] Z. Wang and T. Schultz, Non-Native Spontaneous Speech Recognition through Polyphone Decision Tree Specialization, in *EUROSPEECH*, pp. 1449–1452, 2003.

[52] Janus Recognition Toolkit: http://www.cs.cmu.edu/∼tanja/Lectures/JRTkDoc/.

[53] I. Rogina and A. Waibel, The JANUS Speech Recognizer, in *ARPA SLT Workshop*, pp. 166–169, Morgan Kaufmann, 1995.

[54] H. Soltau, F. Metze, C. Fügen, and A. Waibel, A one-pass decoder based on polymorphic linguistic context assignment, in *ASRU*, pp. 214–217, 2001.

[55] NIST SCLite: http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm.

[56] Statistical Machine Translation Tools: http://www.statmt.org/.

[57] F. J. Och and H. Ney, A Systematic Comparison of Various Statistical Alignment Models, in *Computational Linguistics*, volume 29, pp. 19–51, 2003.

[58] SRI LM: http://www.speech.sri.com/projects/srilm/.

[59] IRSTLM Toolkit: http://hlt.fbk.eu/en/irstlm.

[60] RandLM: http://randlm.sourceforge.net/.

[61] KenLM: http://kheafield.com/code/kenlm/.

[62] N. Bertoldi, B. Haddow, and J.-B. Fouet, Improved Minimum Error Rate Training in Moses, in *The Prague Bulletin of Mathematical Linguistics*, pp. 7–16, 2009.

[63] Z. Mihaylova, T. Schultz, and T. Schlippe, An Architecture of a Telephone-based System for Speech Data Collection, Student thesis 2010.

[64] Festival: http://www.cstr.ed.ac.uk/projects/festival/.

[65] L. M. Tomokiyo and S. Burger, Eliciting Natural Speech from Non-Native Users Collecting Speech Data for LVCSR, in *ACL-IALL Joint Workshop on Computer Mediated Language Assessment and Evaluation in NLP*, 1999.

[66] M. Vondrasek and P. Pollak, Methods for Speech SNR estimation: Evaluation Tool and Analysis of VAD Dependency, in *RADIOENGINEERING*, pp. 6–11, 2005.

[67] Test of English for International Communication (TOEIC): http://www.ets.org/toeic.

[68] L. M. Arslan and J. H. L. Hansen, Language Accent Classification in American English, in *Speech Communication*, 1995.

[69] Amazon Mechanical Turk: https://www.mturk.com/.

[70] R. Fisher, The Use of Multiple Measurements in Taxonomic Problems, in *Annals Eugen*, volume 7, pp. 179–188, 1936.

[71] H. Lang, M. Raab, R. Gruhn, and W. Minker, Comparing Acoustic Model Adaptation Methods for Non-Native Speech, in *International Conference on Acoustics (NAG-DAGA)*, 2009.

[72] Trainable grapheme-to-phoneme converter Sequitur G2P: http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html.

[73] R. Rosenfeld, Two Decades of Statistical Language Modeling: Where do we go from here?, in *IEEE*, volume 88, pp. 1270–1278, 2000.

[74] GlobalPhone website: http://www-2.cs.cmu.edu/~Etanja/GlobalPhone/.

[75] T. Schultz, A. W. Black, S. Badaskar, M. Hornyak, and J. Kominek, SPICE: Web-based Tools for Rapid Language Adaptation in Speech Processing Systems, in *INTERSPEECH*, 2007.

[76] A. W. Black and T. Schultz, Speaker clustering for multilingual synthesis, in *MultiLing*, 2006.

# Index

# List of Figures

# List of Tables