Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN

Masterthesis

Using data augmentation to improve speech recognition for low resourced languages - Amharic

at

Language Technology (LT)

Abgabedatum:	31.01.2020
Name:	Valentin Strauß
E-Mail:	1strauss@informatik.uni-hamburg.de
Studiengang:	M.Sc. Informatik
Matr.Nr.:	6328842
Fachsemester:	7

Erstgutachter:	Prof. Dr. Chris Biemann
Zweitgutachter:	Prof. Dr. Wolfgang Menzel
Betreuung:	Benjamin Milde, Seid Muhie Yimam

Abstract

When it comes to speech recognition for low resourced languages, researchers are confronted with low data problems and the lack of linguistic expertise. Also, there is a minor political and economic interest in developing automated speech recognition systems for these languages. Nevertheless, it is an essential technology when it comes to collecting and preserving individual and cultural information thus it is desirable to develop ASR systems for uncommon languages.

This thesis investigates the influence of different data augmentation strategies on the performance of speech recognition systems for low resourced languages.

With Amharic as target language, the effect of 4 different augmentation methods on Gaussian Mixture Model based and Time Delayed Neural Network based acoustic models observed.

The Amharic language has a rich morphology and thus many different words. In order to handle the problem of out of vocabulary words, a sub word unit language model with graphemes of the Amharic writing system was proposed and the results are compared to word-based language models.

The results showed, that Time Delayed Neural Networks and data augmentation strategies are good approaches to improve the accuracy of speech recognition systems for low resourced languages.

The grapheme-based language model is able to detect out of vocabulary words, but the overall performance of word-based language models is much better.

Content

List of Abbreviations

Abbreviation:	**Explanation:**
AM	Acoustic Model
ANN	Artificial Neural Network
ASR	Automated Speech Recognition
CV	Consonant-Vowel
EM	Expectation Maximization
GMM	Gaussian Mixture Model
GMM	Gaussian Mixture Model based approach
HMM	Hidden Markov Model
HLT	Human Language Technology
LM	Language Model
OOV	Out of Vocabulary
P	Pitch based approach
PDF	Probability Distribution Function
S	Speed-based augmentation strategy
SER	Sentence Error Rate
SyER	Syllable Error Rate
SYL	Syllable-based Language Model approach
T	Tempo-based augmentation strategy
T/P	Tempo + Pitch-based augmentation strategy
TDNN	Time Delayed Neural Network
TDNN	Time Delayed Neural Network based approach
VTLN	Vocal Tract Length Normalization
VTLP	Vocal Tract Length Perturbation
WER	Word Error Rate
WORD	Word-based Language Model approach

List of Figures

List of Tables

1 Introduction

The field of Human Language Technologies (HLT) deals with the processing of human language. Popular languages like English receive more attention in research, since they are more attractive than other, less common languages. On the one hand there is a high political and economic interest in developing speech recognition systems for popular languages, on the other hand there are large datasets of transcribed recorded speech data available for these languages.

For humans, language is the natural means of communication and information exchange. This information can be individual of collective ideas, memories, findings and practices. Language is also a key aspect in cultural identity and empowerment.

Globalization and digitalization connect the people of the world and opens possibilities and reasons to communicate with each other. This may give the opportunity to share and preserve cultural knowledge and information about a low resourced language. But globalization and digitalization also bears the risk of language extinction for low resourced languages which would be a loss for the whole human culture (Besacier et al., 2014).

Languages, where huge datasets are available for research, are called well-resourced languages whereas low resourced languages lack these datasets. Large datasets allow complex model training and lead to a superior performance of speech recognition systems.

For well-resourced popular languages there are well transcribed datasets of high quality, some of them recorded only to support HLT research (Zhang and Glass, 2009).

Nevertheless, the development of speech recognition systems for low resourced languages is an important and challenging task for HLT.

Developing speech recognition systems for low resourced languages may weaken factors that causes an unpopular language to die and thus preserve parts of the human culture.

This master thesis will analyze the problems of developing speech recognition systems for low resourced languages. The main problem when developing automated speech recognition systems for low resourced languages is the lack of data. Therefore, the influence of several data augmentation techniques on different acoustic and language models is investigated for the Amharic language.

This thesis begins with an introductory motivation followed by the research questions. To introduce the topic, there is a general overview of the topic of speech recognition, in which different language and acoustic models are discussed. Chapter 3 then examines relevant research that focuses primarily on speech recognition for the Amharic language and explains augmentation methods. This serves as a transition to the following chapter, in which the approaches and experiments of this thesis are described. The results are then presented in chapter 5 and serve as the basis for the discussion that follows, which ultimately picks up and answers the research questions again. The thesis is concluded by considering open research areas and recommendations for further work.

1.1 Motivation

Automated Speech recognition systems provide a lot of advantages. Since spoken language is the main natural method of human communication, these systems have a broad range of applications. They can help individuals in their everyday life by taking away the task of writing things down or translating a spoken input into a computational command and thus make it easier and more intuitive to operate with computers. This is especially useful for disabled people and can help them to be more independent. For example, people who are physically unable to use a keyboard can use their spoken language to write an e-mail.

Speech recognition systems can also help larger projects to acquire processable data from audio sources. The ability to translate spoken language into plain text can help companies, for example, to predict their customers' wishes by transcribing verbal feedback and making it automatically processable.

A spoken sentence usually contains information with the intention to transfer this information. The technology to record audio data is much older than the attempt to recognize the spoken content of these recordings. According to this, there is a lot of recorded spoken data which contains historical and cultural information. So, another important application of speech recognition is to process recordings of the past to digitize this information. This task will become more and more important because with modern technology it is possible for almost everyone to record situations of their life and make them public. Due to this, there is a lot of data which may contain useful information but without the ability of automatically transcribing this data, remains inoperative.

As mentioned before, modern technologies provide the possibility for individuals to store and share knowledge across the world. For low resourced languages, this may provide the opportunity to preserve and share cultural knowledge and. As Besacier et al. (2014) stated, globalization and digitalization also bears the risk for low resourced languages to be repressed by other more popular languages, which would be a loss for the whole human culture.

Language can be a barrier when it comes to human-to-human communication, so developing speech recognition systems for low resourced languages may weaken factors that causes a low resourced language to suffer and thus preserve parts of the human culture.

For the human brain the task of recognizing a spoken sentence seems fairly easy. Based on their experience, humans have an enormous memory of related knowledge to decode perceived sentences. We can use this knowledge to verify a spoken sentence in terms of grammatical correctness and furthermore analyze the sense of the utterance and check the pragmatical correctness. When it comes to automated speech recognition, many challenges must be overcome to recognize the content of an utterance correctly.

According to Isern and Fort (2014), language evolution is a slow process. It can take over a thousand years for any language to develop to other languages. As the world gets more globally connected, the language extinction rate fastens. As Figure 1 illustrates, from

approximately 7000 languages, only 4% are spoken by 96% of the world's population and since this is a long tail distribution, about 25% have less than 1000 speakers.

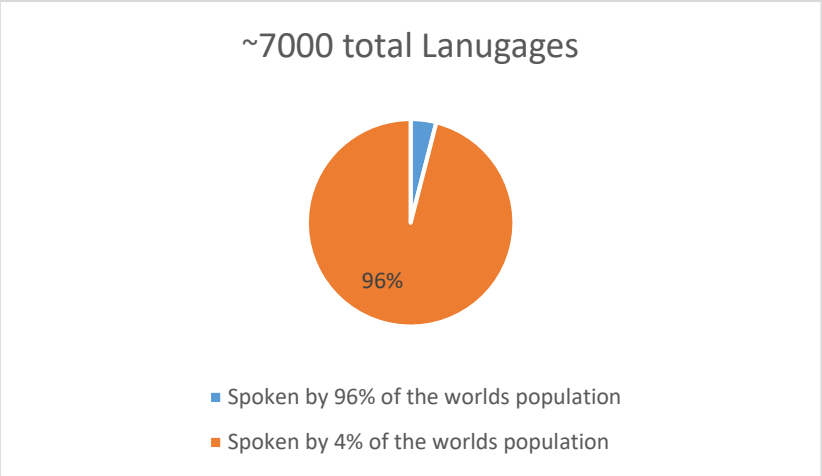

Figure 1: Illustration of the proportion between languages of the world and humans that speak these languages

According to Besacier et al. (2014) the pressure on a language is crucial for its survival. Languages with high pressure and few speakers have low chances of survival. The pressure can have natural causes like earthquakes or other disasters that can wipe out cultures and languages but the main pressure comes from other surrounding languages. The dominance of another language can result in cultural convergence. Speakers of the endangered languages have an interest to adapt to the dominant culture due to potential economic benefits and especially the young generations are affected (Isern and Fort, 2014).

There are already well-developed language processing systems for several popular languages. It is hard to tell how many languages exist but it is estimated, that there are about 7000 languages in the world (Isern and Fort, 2014; Besacier et al., 2014) and only a fraction of them are suitable for language processing systems due to a lack of data. Therefore, the main focus in language processing research is on languages where much data is available for training and testing the system. Another factor is whether it is in the political or economic interest to develop such systems for a specific language. As a result,

for many languages in developing countries, research on language processing systems receives very little attention (Besacier et al., 2014).

1.2 Research Question

The main research question of this thesis is:

What kind of model can be recommended for automated speech recognition when working with low resourced languages?

To answer this question, the following questions must be answered first:

1. Is it possible to improve the performance of automated speech recognition systems for low resourced languages with data augmentation techniques?

- **1.1:** Which data augmentation techniques are suitable for low resourced languages?
- **1.2:** Which augmentation strategy works best?

2. Is it possible to improve speech recognition for low resourced languages using different acoustic and language models?

- **2.1:** Which acoustic model is best suited for the task of speech recognition for low resourced languages?
- **2.2:** Which language model is best suited for the task of speech recognition for low resourced languages?

3. What is the best combination of augmentation strategy, acoustic model and language model?

4. Since we use Amharic as a low resourced language target, can the concluded model be transferred to other low resourced languages?

1.3 Low resourced languages

As Besacier et al. (2014) stated, an low resourced language refers to a language that fulfills some, but not necessarily all, of the following aspects:

- The first aspect is the lack of a stable orthography or writing system. Since Automated Speech Recognition deals with the processing of spoken language to plain text, it is a precondition for a sufficient recognizer, that the language has a stable orthography and writing system. Without such, the results of the standard training and testing phase are doomed since the recognition may be correct but the related test phrase is spelled in another way. Models that deal with these kinds of languages need additional expert knowledge to overcome this barrier.
- Another aspect is the rare occurrence of this language on the web in written as well as in recorded format. Since the best performing of machine learning algorithms generally depend on the available amount of data for training, researchers use the web to expand their corpora with additional data to improve the results of speech recognizer (Zhu and Rosenfeld, 2001).
- The lack of linguistic expertise and the lack of electronic data available for speech processing such as mono- or bi-lingual corpora, dictionaries as well as pronunciation dictionaries and transcribed speech are also factors, that define if a language is low resourced.

As mentioned before, globalization and digitalization bear risks and opportunities for low resourced languages. On the one hand, it connects people and opens possibilities and reasons to share and preserve cultural knowledge and information about a low resourced language. On the other hand, more popular and dominant languages have a larger influence on low resourced languages and thus the risk of a low resourced languages to suffer rises (Besacier et al., 2014).

Language can be a barrier when it comes to human-to-human communication, so developing speech recognition systems for low resourced languages may alleviate factors that causes an unpopular language to extinct and thus preserve parts of the human culture.

Vries et al. (2014) stated, that the future economic sustainability of a language can be strengthened with automated speech recognition systems as they can support domains

such as education, health-information services, information-access, and government services and agriculture.

It is hard to say how many languages exist but it is estimated, that there are about 6000-7000 languages in the world (Isern and Fort, 2014; Besacier et al., 2014) and only a fraction of them are suitable for language processing systems due to lack of data. Vries et al. (2014) stated, that there are only about 30 languages where the amount of data available for machine learning is suitable. Therefore, the main focus in language processing research is on languages where much data is available for training and testing the system. Another factor is whether it is in the political or economic interest to develop such systems for a specific language. As a result, for many languages in developing countries, research on language processing systems receives very little attention(Besacier et al., 2014).

But as already motivated, there are good reasons to develop speech recognition systems for the low resourced languages too.

Since the performance of speech recognition systems strongly depends on the size of the dataset, the main problem of rare languages is the available amount of data of for training and testing the models.

In different cultural environments the human race has developed different kinds of writing systems. These systems may differ in the shape of symbols, as well as the meaning of a single character.

As Killer et al. (2003) stated, alphabets are very common and represent consonants and vowels. The most common alphabet is the roman alphabet which is used by most European nations, as well as most of the African American and Oceanian nations. Even some Asian nations use the roman alphabet. Although these nations share the same alphabet many languages have modified the roman alphabet with extra letters. Figure 2 shows an example of the popular roman script.

Figure 2: Example of the roman alphabet

Similar to alphabets are abjads or consonant alphabets. An abjad represents only consonants (or sometimes even a few vowels) and are usually right-to-left written. Some written languages, like Arabic, can be used as alphabet and as abjad.
In contrast to alphabets, a syllabic writing system consists of symbols that represent either a single vowel. Figure 3 shows an example of the Amharic syllabic writing system.

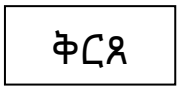

Figure 3: Example of the Amharic script

A syllabary is a special form of a syllabic alphabet where each syllable of the language is represented by its own symbol. An example of syllabary is the Japanese Hiragana.
Logographic writing systems usually have the largest set of symbols. Each symbol represents not only a sound but also a meaning. Theoretically, there is no upper limit for the number of distinct symbols. Chinese is for example a logographic language.
There are some alternative writing systems that do not fit in the classification above. These scripts are invented for example in books, movies or computer games (Killer et al., 2003).

1.3.1 Amharic

Amharic is a South Semitic Ethiopian language and belongs to the Afro-Asian language family.
According to Kramer (2009), there are three main branches within the semitic languages: East (Akkadian, Old Babylonian, etc.), Central (Hebrew, Aramaic, Arabic, etc.) and South (South Arab, Ethiopian). Amharic is a member of the Ethiosemitic languages and is classified within Ethiosemitic along with Argobba, Harari and the languages of East

Gurage for example Southern Transversal. Other Ethiopian languages include Tigre and Tigrinya, both of which are spoken in northern parts of Ethiopia, and some minority languages spoken in Ethiopia and Eritrea, such as Gafat, Mesmes, and Inor.

Amharic is the semitic language group that has the largest number of speakers after Arabic (Abate et al., 2005) and is the national language of Ethiopia, which is taught in schools and used in national newspapers and government publications. In Ethiopia, about eighty languages are spoken, including thirteen other Semitic languages, many Cushitic languages (including Oromo, Sidamo and Afar), many Omotic languages and several languages from the Nilo-Saharan family. According to Sarah Adam (2019), there are about 30 million Amharic native speakers.

There are five different dialects for Amharic, based on regions: Addis Ababa, Gojjam, Gondor, Wollo and Menz, where Addis Ababa is the most spoken one (Abate et al., 2005).

The Amharic pronunciation system is characterized by a homogeneous phonology distinguishing between 234 distinct Consonant-Vowel (CV) syllables.

Like other languages, there are some unique sounds that cannot be found in other languages for example some click-like characters (Abate et al., 2005).

The Amharic language consists of seven vowels shown in Table 1 and thirty-one consonants. As Figure 4 shows, the consonants can be generally classified in five categories: stops, fricatives, nasals, liquids and semi-vowels (Abate and Menzel, 2007).

Manner of Art/n	Voicing	Place of Articulation				
		Lab	Dent	Pal	Vel	Glo
Stops	Voiceless	ፕ[p]	ት[t]	ች[tʃ]	ክ[k]	አ[ʔ]
	Voiced	በ[b]	ድ[d]	ጅ[dʒ]	ግ[g]	
	Glottalized	ጵ[pʼ]	ጥ[tʼ]	ጭ[tʃʼ]	ቅ[q]	
	Rounded				ኩ [k^w] ጉ[g^w] ቁ[q^w]	
Fricatives	Voiceless	ፍ[f]	ስ[s]	ሽ[ʃ]		ሀ[h]
	Voiced		ዝ[z]	ዥ[ʒ]		
	Glottalized		ጽ[sʼ]			
	Rounded					ኍ[h^w]
Nasals	Voiced	ም[m]	ን[n]	ኝ[ɲ]		
Liquids	Voiced		ል[l] ር[r]			
Semi vowels	Voiced	ው[w]			ይ[j]	

Figure 4: Categories of Amharic consonants. Lab=Labial; Den=Dental; Pal=Palatal; Vel=Velar; Glo=Glottal (Abate and Menzel, 2007)

	front	center	back
high	ሊ [i]	ል [ɨ]	ሉ [u]
mid	ሌ [e]	ለ [ə]	ሎ [o]
low		ላ [a]	

Table 1: Categories of Amharic vowels (Abate and Menzel, 2007)

Abate and Menzel (2007) state that each of the five dialects use the same writing system. According to Leslau (2000) there is in general a one-to-one mapping between the spoken syllables and the grapheme symbols. Each grapheme represents a combination between a consonant and a vowel (CV-syllable) or a single vowel.

The Amharic alphabet consist of 276 different characters, but there are redundant graphemes for the same syllable, so in total there are 234 distinct syllables.

Amharic is a morphologically rich language because it has an inflectional and derivational morphology (Abate and Menzel, 2007). This has the consequence that there

are many individual words in a dataset. For automated speech recognition, this is a serious problem, because there is a high chance that the model is confronted with unseen words, so called out of vocabulary (OOV) words. The problems of OOVs and approaches to be able to deal with them will be explained in detail in section 2.3.

2 Speech Recognition Basics

As Jurafsky and Martin (2014) stated, an automated speech recognition system deals with the task of transcribing a spoken utterance into a plain text representation. The problem of speech recognition can be described as:

$$\hat{W} = arg\ \max_{w} p(W \mid A)$$

Where $\hat{W}$ is the word-sequence that maximizes the conditional probability and $p(W|A)$ is the probability that the word sequence W was observed given the acoustic evidence A.

This can be transformed using Bayes Theorem:

$$\hat{W} = arg\ \max_{w} \frac{p(A \mid W)p(W)}{p(A)}$$

Where $p(A)$ is the probability, that the acoustic evidence A has occurred, $p(A|W)$ is the probability, that A is observed knowing, that the speaker spoken W and $p(W)$ is the probability of the word-sequence W.

The probability $p(A)$ will not change for a given input and thus it can be disregarded in this equation, so the task of speech recognition can be formulated as finding $\hat{W}$ where $\hat{W}$ is:

$$\hat{W} = arg\ \max_{w} p(A \mid W)p(W)$$

In Figure 5 Jurafsky and Martin (2014) illustrated the basic speech recognition process with its main components which also shows, how the different probabilities are used to recognize a sentence.

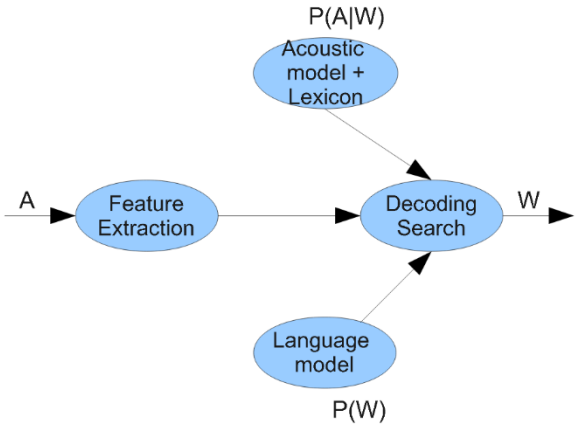

Figure 5: The speech recognition process as illustrated by Jurafsky and Martin (2014) with the acoustic model P(A|W), and the language model P(W)

The feature extraction gets an audio signal as input and transforms it into spectral features. The acoustic model computes the probability *p(A|W)* using the pronunciation dictionary whereas the language model computes the probability of the sentence W. The decoder uses the output of the feature extraction, the acoustic model and the language model to search for the sentence that maximizes the product of *p(A|W)* and *p(W)*.

The difficulties of ASR-systems lie in aligning the data (for example where does a word or a phone start and end), the complexity of the data (how many different words are there and how many different combinations of all those words are possible), the variability of the speakers (women compared to men have a higher fundamental frequency; or microphones, telephones limit the bandwidth, etc.), the ambiguity of words (two vs. too) or word boundaries (interface vs. in her face), syntax (he saw the grand canyon flying to New York) and ambiguities (time flies like an arrow). Automatic Speech Recognition started with speaker-dependent single word recognizers that processed only a small amount of words in quiet surroundings.

Today's focus lies on system development for spontaneous or colloquial speech with a noisy background as it can be found in a car. The systems must be able to adapt to new situations quickly and optimally.

Zhu and Rosenfeld (2001) stated, that in general, there are two kinds of methods when it comes to improving the performance of a speech recognition system.

The first kind focuses on improving the estimation methods for fixed datasets to get better results. Examples for these techniques are among others: n-gram length, smoothing techniques, vocabulary clusters, decision trees and probabilistic context free grammars etc.

The second method focuses on acquiring more data that can be useful for training the model. Examples are language independent acoustic models, cross-language transfer and language-adaption, bootstrapping and using tools to collect data from the web (Vries et al., 2014).

2.1 Acoustic Model

The Acoustic Model (AM) models the conditional probability $P(A|W)$ of an acoustic signal X given a word sequence W. Therefore we need an appropriate design that approximates the problem. A word can sound very different depending on coarticulation effects, speaker dependent pronunciation variants or characteristics of the transmission channel. Since it is infeasible to model $P(X|W)$ for each word sequence (there are way too many possible word combinations), smaller units are modeled.

The fragmentation of words into smaller units introduces a few other problems. First of all, a pronunciation dictionary is required to split the words into the subunits. Secondly a time alignment is needed. The beginning and ending of subunits have to be found first. There are various other problems in automatic speech recognition. There are coarticulation effects at word transitions (a word be pronounced very differently depending on the context). For example, American English has a lot of coarticulation effects.

Most known continuous speech recognition systems at time are based on the idea of Hidden Markov Models (HMMs). Speech can be seen as a stochastic process. This captures that the same phoneme can be pronounced very individually by various people and even the same person pronounces phonemes differently.

2.1.1 Suitable Units for Speech Recognition

In the beginnings of speech recognition, when the task was to recognize and identify single words, usually the base recognition units have been words or morphemes as subunits of words (Killer et al., 2003). The problem with using whole words as recognition units is, that - since every word has to be trained separately - information in form of parameters cannot be shared among different words and the dataset used for training must be very large to cover all possible words sufficiently (Thangarajan et al., 2008). Another problem with word-based approaches is the memory requirement, which grows linear with the number of recognizable units. The number of different words for languages, especially morphological rich ones, can grow very large for spontaneous speech. Since this thesis deals with low resourced languages, the datasets can be considered comparably small, so there is a need for another solution for recognizable units.

Words can be segmented into sub-word units called morphemes. Morphemes are the smallest meaningful units in a language. Some morphemes may be complete words and thus are called a root, for example '*go*' is a root-morpheme. Other morphemes are prefixes and suffixes with can be combined with a root-morpheme to build a word for example the word '*ongoing*' consists of three morphemes: '*on+go+ing*'.
Morphemes are well fitted for a single word recognizer. But with morphemes we have a similar problem as with words as recognition units. Although the total number of morphemes is smaller than the total number of words for a language, the amount of possible words and word combinations is so large, that it is laborious to write down all possible morphemes and it is nearly impossible to find enough training data for each such recognition-unit (Killer et al., 2003).

Another way to split up words is to decompose them into their syllables. Since a syllable is a combination of phones, coarticulation effects between phones are already covered to a certain degree. The total number of syllables is way below the number of words or morphemes, but the problems mentioned above may still arise.

An even smaller unit than the syllables are the phonemes and thus the number of different units also decreases. Usually, there are in between 30 to 50 phonemes and sub-phonemes (Killer et al., 2003).

Because of the low number of phonemes, it is possible to look at them with respect to their context and consider also their left and right neighbor. A phoneme-based recognition unit that also considers the left and right context is called a triphone and is widely used in state-of-the-art speech recognition systems. Despite its name, a triphone often uses phonemes instead of phones. A phone represents a single sound (for example a 'r' can be pronounced rolled and unrolled which results in two different phones), whereas a phoneme combines phones with no difference in meaning, so the words do not change when using different phones of the same phoneme to pronounce the words. If the neighborhood of the phoneme is unspecified it is called a polyphone. The larger the context of a polyphone, the larger grows the total number of different units and like for the higher-level recognition units the trainability may suffer. The total number of possible polyphones depends on the set of phones, the dictionary and, if available, grammatical constrains.

Young et al. (1994) stated, that when including cross-word triphones, the total number of possible triphones will result in a large number of states in the models and many triphone occurrences in the test-data will have a close to zero probability because there are very few occurrences in the training-data. Furthermore, context dependent models generalize less the wider the context neighborhood (Killer et al., 2003). One way to deal with this problem is to use phoneme clusters instead of definite phonemes as context. Therefore, Young et al. (1994) proposed a model which uses state tying. Contextually equivalent HMM states are found by using phonetic decision trees. Therefore, clustering algorithms are used to merge the various contexts of a phoneme together. The clustering procedure can be based on a decision tree which clusters states of the same phoneme in occurring different contexts. A phonetic decision tree is a binary tree where a question is asked in each node. The questions are phone-related and ask for information about the left and the right context. The question always has the form: 'Is the left or right phone a member of the set X' where X can be used to model different phonetic categories such

as nasal, stops, fricatives etc. or even individual phones. An example question may be: 'Is the phone on the left of the current phone nasal?' (Young et al., 1994). Usually the questions that split the branches of the decision tree are linguistically motivated and formulated by an expert (Killer et al., 2003). Questions of the decision trees are scored based on their entropy loss and those questions that have the highest score are applied next to split the node into two child nodes.

Clustering the nodes in the context brings another advantage. It is quite conceivable that triphones that were not seen in the training phase appear in the test data set, especially when the model uses cross-word dependencies (Young et al., 1994). When clustering the context of a polyphone with decision trees, unseen triphones can be synthesized by constructing the triphones based on the tied state association for that triphone's context, which is the leaf node of the decision tree.

Figure 6 shows an example of a phonetic decision tree.

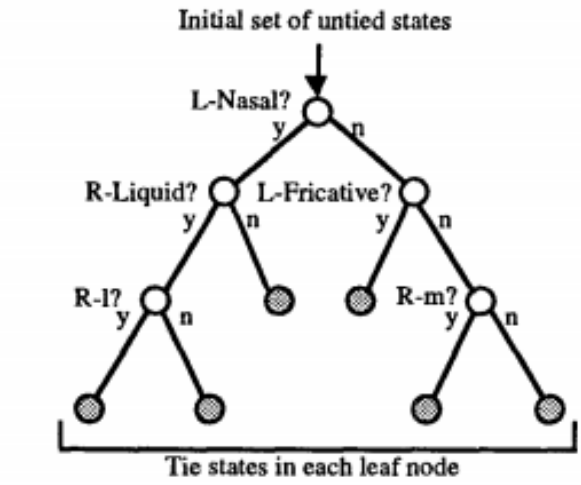

Figure 6: Example of a phonetic decision tree (Young et al., 1994)

Young et al. (1994) proposed a method to create decision trees using the top-down sequential optimization procedure.

In the first step all the states are located in the root node of the tree. The log likelihood is calculated on the assumption, that all the states are tied in the root node. In the second step, the states are assigned to two child nodes by finding the best question which

separates the states with a maximum increase in log likelihood. This step will be repeated for the split which results in the largest log likelihood gain until the improvement fells under a certain threshold. To ensure that all of the nodes have enough data to be sufficiently covered, there is a minimum threshold for states in a node.

The performance of automated speech recognition systems can be improved when using context dependent models. Choosing the right context is essential though and depends on the language as well as on the amount of available training data.

2.1.2 Gaussian Mixture Model

A widely used method for modeling the acoustic probabilities is the Gaussian Mixture Model as described by (Reynolds, 2009), since it is a good way of approximating probability distributions.

The GMM models the probability of an observation, given a class, with a mixture of Gaussian. For each feature in the input vector the GMM models the mean and variance of the probability distribution function PDF for each acoustic unit. The result is a weighted sum of several Gaussian probability distribution functions for each acoustic unit.

Due to computational complexity instead of using the full co-variance matrix, often the diagonal co-variance matrix is used which only contains the variances of each feature.

This means that in practice we are keeping only a single separate mean and variance for each feature in the feature vector.

Each acoustic unit is modeled by a mixture of Gaussian and the GMM is trained by the Expectation Maximization (EM) algorithm:

- M: For each utterance, taking the words and align each acoustic unit to the corresponding acoustic feature vectors using the maximum likelihood. This way, the acoustic frames get labeled.

- E: Given the alignment, the mean and variance (μ,σ) for each acoustic unit are estimated.

The M and E step are repeated until the improvement on the training data is under a certain threshold.

A clear advantage of GMMs is, that this method is highly parallelizable. Having a model, the alignment step can be done for each utterance in parallel. After the alignment step, the information about each acoustic unit can be collected and every single mixture model can be trained in parallel for a specific acoustic unit.

The disadvantage of GMMs is that during the alignment step each frame is aligned to exactly one corresponding acoustic unit and in the M step the frames train only the specific model of the aligned acoustic unit, so there is no generalization.

2.1.3 Time Delay Neural Networks

Waibel et al. (1989) presented a method called Time Delay Neural Network (TDNN) to model the acoustic properties of phonemes. To ensure, that the network is able to learn complex nonlinear decision surfaces, the TDNN is a feed-forward neural network with multiple layers and links between the units in each of these layers.

The network aims to represent the relation of events based on different time steps with no temporal per-alignment of acoustic units during training.

Usually for neural networks, the basic unit computes the weighted sum of its inputs and passes the result through an activation function (for example sigmoid or threshold). However, the TDNN is slightly modified, since it uses multiple time-steps as input.

Assuming the model uses N=2 delays (D1, …, DN) with J=16-dimensional feature input vectors, units in the first hidden layer are receiving 48 weighted input connections. This way the network has the ability to relate the input of the current frame to events in the past.

Figure 7 shows an example of a 3-layer TDNN for phoneme recognition. The feature vectors used consists of 16 normalized Mel Frequency Cepstral Coefficients (MFCC). The second layer is fully connected to a 3-frame window from the input layer. It consists of

eight time-delayed hidden units, each connected with the 16 corresponding inputs of the current frame and the 32 inputs of the delayed frames.

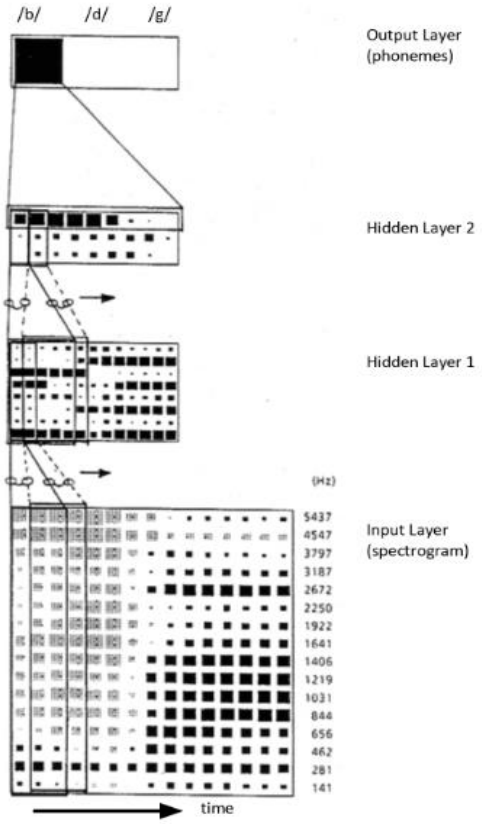

Figure 7: The TDNN as presented by (Waibel et al., 1989) for phoneme recognition

The second hidden layer consists of three hidden units and uses a 5-frame window of the one hidden layer, so the number of incoming connections for each unit in this layer is 40. The larger window of frames is motivated by the assumption, that higher level units should consider a larger period of time when making decisions.

By summing up the evidence from each of the three nodes in the second hidden layer and applying a sigmoid function, the resulting phoneme can be obtained from the nodes in the output layer.

In conclusion each unit of the TDNN is able to encode temporal relations based on the range of the delays, where higher level units consider a larger period of time.

Training of an TDNN is done via error-back-propagation, where the weights of the TDNN are iteratively adjusted, using labeled training data, so that the error for the training data gets minimized (Jurafsky and Martin, 2014).

Based on the assumption, that there are large context overlaps between activations computed at neighboring frames, Peddinti et al. (2015) proposed a TDNN that uses sub-sampling by allowing gaps between the delays in each layer. While the first hidden layer is fully connected to a 5-frame window from the input layer, the higher-level units only use the output of two nodes from the previous layer.

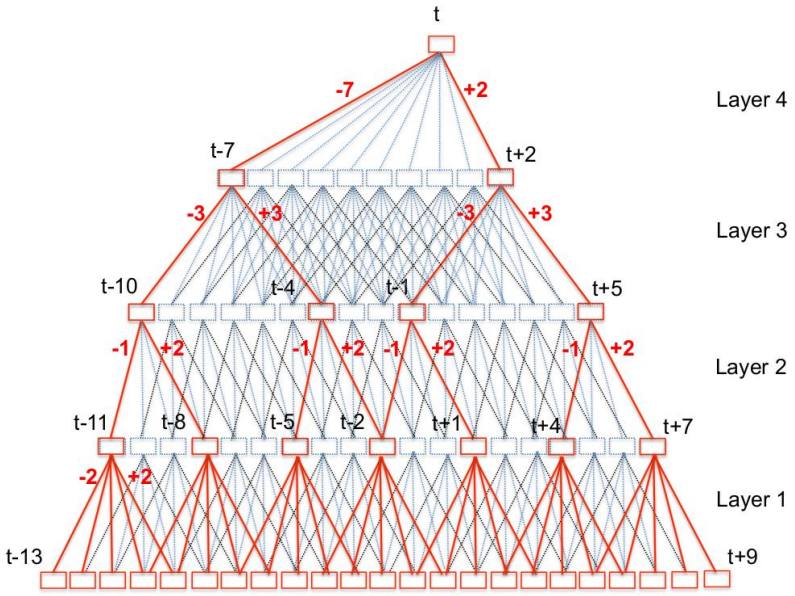

Figure 8: The architecture of a TDNN that uses sub-sampling (red connections) and no sub-sampling (red and blue connections) as illustrated by (Peddinti et al., 2015)

The architecture of a TDNN with and without sub-sampling can be seen in Figure 8. This strongly reduces the number of computations during training and the process can be done in a fifth of the time compared to no sub-sampling training.

2.2 Pronunciation Dictionary

The pronunciation dictionary is one the core components of an automated speech recognition system. In the pronunciation dictionary, each word is divided into its subunits. For speech recognition systems with very large vocabularies, it is a bad idea to

use whole words as basis for unit-classification, because the number of states is so large that it is nearly impossible to get enough training data to cover the states sufficiently. Furthermore, there is a high risk of being confronted with untrained words in the training data, these words have no chance of being recognized correctly. With the pronunciation dictionary, the number of recognizable states is highly reduced. Instead of using the whole word as recognition unit, the words are represented in subunits. This method is more robust, because the subunits occur far more often and are shared between the different words in the lexicon. This makes it even possible to deal with unseen words, because they can be decomposed into their subunits. Usually, words are decomposed into their phoneme-representation and thus the pronunciation dictionary defines the pronunciation of the words (Killer et al., 2003).

The pronunciation dictionary can have a strong impact on the results of the speech recognition system and the choice of subunits is an important task. If similar sounds, for example in English 'D' and 'T', would be represented by the same subunit the recognizer could not differ between 'BAD' and 'BAT' (Singh et al., 2002). Because of the importance of the pronunciation dictionary, usually language experts are given the task to manually create these phoneme-based dictionaries. The problem with rare languages is, that the interest in developing speech recognition system for these languages is comparably low and thus there is low effort in creating these dictionaries (Besacier et al., 2014). So, in order to create applicable pronunciation dictionaries other methods are needed.

There has been effort in automatically creating pronunciation dictionaries and researchers have come up with different solutions. Most approaches use segmentation and clustering methods to treat the problem of identifying sub-word units (Singh et al., 2002). These approaches heavily rely on the available training data and thus may be not suited for rare languages.

It is also possible to use trained models of other languages to build the pronunciation dictionary. This method uses multiple phoneme recognizers for different languages and uses them to decode an acoustic frame for each of these languages. A voting mechanism then decides witch hypothesis is the best fitting one. Experiments however showed, that this method produces rather poor results (Stüker et al.).

Fukada and Sagisaka (1997) obtained good experimental results by using a pronunciation dictionary for spontaneous speech. Spontaneous speech differs from read speech in that it varies more in terms of pronunciation. A phone recognizer and its results were used to train multi-layer perceptron neural networks. The goal was to predict the correct pronunciation from a symbol sequence. In the next step, these networks were used to generate the pronunciation dictionary.

A grapheme-based pronunciation dictionary divides each word into its single characters and uses these as recognition units. This approach may have some shortages to manually created pronunciation dictionaries and the results are expected to be not as good, but on the other hand, a grapheme-based pronunciation dictionary can be created fast without expert knowledge. But this can only work if the words of a language are represented in multiple graphemes. Chinese Hanzi (a logographic script system) for example uses one unique grapheme for each word and thus the words cannot be divided into multiple grapheme-based subunits (Killer et al., 2003). When dealing with writing systems other than the roman one, it is possible to convert the written text into the roman writing system. This method is called romanization and makes it possible to perform grapheme-based language processing even for logographic languages.

Killer et al. (2003) showed, that the performance of grapheme-based recognizers in four different languages (English, Spanish, German and Swedish) are not as good as the ones with phoneme-based recognizers but stated, that for some languages the performance of grapheme based recognizers may be close to phoneme based recognizer. In conclusion, the performance of grapheme-based recognizer is strongly dependent on the language.

2.3 Language Modeling

A Language Model (LM) describes how sentences of a language are structured. Therefore, it models the possible word transitions and determine how words can be combined to word sequences which result in whole sentences.

The LM is one of the key components in an automated speech recognition system and the performance strongly depends on how well the it models the actual structure of a language.

One way to deal with this task is to create grammars. A grammar basically consists of a dictionary with all possible words and word-classes and hand-written rules about how to build a phrase and how phrases can be combined to from a sentence.

There are different kinds of grammars like context free grammars or unification grammar (Jurafsky and Martin, 2014).

This approach may be well suited for tasks with a small dictionary and simple rules for sentence construction but as soon as it comes to recognizing free speech, the effort of creating such models increases dramatically.

To create a grammar a linguistic expert with extensive understanding of the target language is needed. For rare languages, like Amharic, it is hard to create a sufficiently representative grammar due to the lack of linguistic expertise.

Another way to create language models is the statistical approach. These models have the major advantage that no expert knowledge is needed, since these statistical language models are corpus based.

Statistical LMs have the basic assumption, that the probability of a word depends on the previous sequence of words.

Given a word sequence W = $w_1, w_2, \ldots, w_q$ the probability $p(W)$ of this sequence can be calculated by:

$$p(W) = p(w_1)\, p(w_2 \mid w_1),\, p(w_3 \mid w_1 w_2),\, \ldots,\, p(w_q \mid w_1 w_2 \ldots w_{q-1})$$

or:

$$p(W) = \prod_{i=1}^{q} p(w_i \mid w_1, \ldots, w_{i-1})$$

Using the whole word history however is not applicable in practice. The longer a sequence gets, the less is its probability to occur in a dataset. This makes it hard to estimate good parameters especially for free speech since most sequences will only occur a few times and others might not occur at all in the training data.

A solution to overcome this problem is to use n-gram models (Jurafsky and Martin, 2014). This technique is based on the Markov assumption, which indicates that the probability of an event in the future can be predicted by only looking at some of the previous words.

So instead of looking at the entire sequence of words, an n-gram language model uses the previous n – 1 words to estimate the probability of a word in a sequence.

For example, to approximate the probability of a word with a bi-gram model (n = 2), only the immediate predecessor word is taken into account.

The probabilities in the n-gram LM are computed using maximum likelihood estimation, which basically counts all sequences of length n in the corpus and determines the probability for each sequence based on the total number of occurrences of this sequence normalized by the total number of all sequences, so that the resulting probability lies between 0 and 1:

$$p(w_q \mid w_{q-N+1}, \ldots, w_{q-1}) = \frac{C(w_q \mid w_{q-N+1}, \ldots, w_q)}{C(w_q \mid w_{q-N+1}, \ldots, w_{q-1})}$$

The complexity of the model increases with higher ordered n-gram models. This effect can be shown by a simple example. Assuming we have a small dictionary with 100 different words, the number of possible uni-grams (n = 1) is the total number of words, since we do not take any previous words into account. When we use a bi-gram model, the number of n-grams increases by a factor of 100 since (theoretically) each word can be combined with every other word to compute the dependent probability. Any next higher order would increase the possible number of n-grams again by a factor of 100. Since the number of n-grams represents the number of search states in the model and thus have a high impact on the computation time.

When increasing the number of states, the coverage for each of these states gets lower. This makes it especially harder to estimate the probabilities of low frequent sequences. That is why usually the order of n-grams is usually not higher than 4.

Another problem with higher ordered n-grams is the data sparsity. The best-case scenario is, that every n-gram occurs several times in the data-set, so that it is possible to sufficiently estimate the probabilities. As mentioned before, when the order of the n-gram model increases, the number of total n-grams also strongly increases and thus each single n-gram occurs less often. This may lead to worse results since the probabilities cannot be estimated adequately.

Since the probability-distribution strongly depends on the corpus and its context, this method benefits from large training sets, but even with a large training set it is still probable, that possible sequences of words do not occur in the training examples. To handle the problem of unknown sequences of words, smoothing techniques can be used. This way the unseen word sequences get a low probability above zero and thus the LM has the ability of detecting those sequences.

Especially for morphological rich languages there is even a high risk, that individual words do not occur in the corpus which is called the Out-of-Vocabulary (OOV) problem (Tachbelie, 2010). This may lead to a negative impact on the performance of the speech recognition system.

2.4 Hidden Markov Models

As mentioned in Section 2.1.1 each spoken word w of an input sequence is partitioned into a sequence of K_w basic sound units called *base phones*. The resulting sequence is called the pronunciation of the spoken word $\boldsymbol{q}_{1:K_w}^{(w)} = q_1, \dots, q_{K_w}$.

Figure 9 shows, how the *base phones* are represented as continuous density Hidden Markov Models (HMMs) with $\{a_{ij}\}$ as transition probability and $\{b_j()\}$ as output observation distribution (Gales et al., 2008).

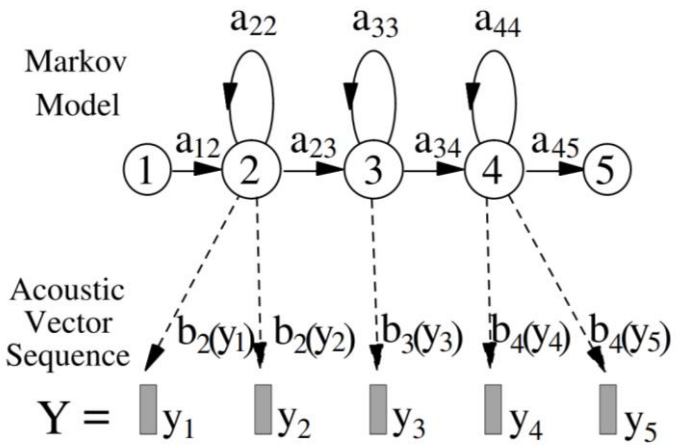

Figure 9: HMM-based phone model (Gales et al., 2008)

At each step, an HMM makes a transition from the current state s_i to the next state s_j, where the transition and emission probabilities are considered when deciding which next state will be chosen.

This way, the conditional independence assumptions of an HMM are ensured (Gales et al., 2008):

The first conditional independence assumption is, that states are only conditionally depend on the preceding state. In reverse this means that a state only influences the next state and no other.

The second assumption states, that observations are not dependent on each other but only on the coherent state and its output observation distribution.

Jurafsky and Martin (2014) summarized the parameters of a Hidden Markov Model as:

- The **states:** $\boldsymbol{q}_{1:K_w}^{(w)} = q_1, \ldots, q_{K_w}$
- The **transition probabilities**: The transition probabilities $A = a_{01}, a_{02}, \ldots, a_{n1}, \ldots, a_{nn}$. The set can be seen as a **transition probability matrix** where each a_{ij} is the transition probability from state i to state j.
- The **observation likelihood:** Each state i may result in multiple observations y_t, so the set of observation likelihoods is $Y = b_i(y_t)$, where $b_i(y_t)$may have any value from 0.0 to 1.0 and expresses the probability of an observation y_t being generated from a state i.

- The **initial distribution**: An HMM needs to start in a specific state, so $\boldsymbol{\pi}$ is the initial probability distribution over all states. The probability, that state i is an initial state and the HMM will start in i is given by π_i. There may be states *j* where $\pi_j = 0$ which indicates, that these states do not occur at the beginning of words.

3 Related Work

This chapter will give an overview of researches in the past that are related to the topic of this thesis. In section 3.1 different data augmentation techniques are explained and discussed of they are applicable for low resourced languages. Section 3.2 provides information about speech recognition for Amharic and states approaches and results of researches in the past.

3.1 Data Augmentation

As mentioned before, there are often no well-developed transcribed datasets for low resourced languages. This circumstance makes is hard to estimate robust parameters for the speech recognition system (Ragni et al., 2014). One way to overcome this shortage is to transform the input in a way that does not change the label or transcription of the data. This method - called data augmentation - first was used for visual object recognition, where the input was horizontally or vertically transformed, rescaled or local image distortions where added to obtain more training examples from the given data (LeCun et al., 1998). In speech recognition, data augmentation is a common method to increase the quantity of your training examples (Ko et al., 2017) and it is especially useful for small datasets where the number of examples is low.

Since data augmentation methods do not really create new samples but rather create a new perspective on the available data, the performance of a speech recognition system would benefit more from not previously seen utterances, but due to the limited resources problem, data augmentation offers a suitable attempt to handle this problem for low resourced languages.

One way to augment speech datasets is to alter the audio examples with the Vocal Tract Length Perturbation (VTLP) method (Jaitly and Hinton, 2013). VTLP maps the frequency of an audio signal f to a new frequency f' by applying a warp factor α to the frequency of the input data:

$$f' = \begin{cases} f\alpha, & x \leq F_{hi}\min(\alpha, 1)/\alpha \\ \frac{S}{2} - \frac{S/2 - F_{hi}\min(\alpha, 1)}{S/2 - \frac{F_{hi}\min(\alpha, 1)}{\alpha}}, & otherwise \end{cases}$$

With the sampling frequency S and a boundary frequency F_{hi} that covers the significant formants.

Jaitly and Hinton (2013) stated, that the warp factor α is usually chosen between 0.8 and 1.2 because a larger range tends to create unmanageable distortions.

Vocal Tract Length Normalization (VTLN) in speech recognition is used to reduce the variability between different speakers that results from different lengths of the speaker's vocal tract. Based on the VTLP method, Jaitly and Hinton (2013) proposed an advanced method to improve data augmentation for speech recognition systems. They showed, that transforming inputs by using VTLN to generate a larger augmented training base can increase the recognition accuracy by up to 1.1% on an English speech recognition task. Jaitly and Hinton (2013) used this method to augment an acoustic dataset by randomly generating a wrap factor for each utterance in training to generate new utterances.

Tempo perturbation as mentioned by Ko et al. (2015) is a method, that modifies the tempo rate but ensures, that the signal keeps its original pitch level and spectral envelope. By always combining two frames into one at a certain distance in the input signal, it is possible to shorten the signal. When extending the signal, additional frames are inserted at a certain distance over the whole signal which can be synthesized by the neighboring frames. By applying this method with modification factors α of 1.1 and 0.9 it is possible to generate a training set which is three times as large as the original one.

Ko et al. (2015) also proposed an approach to augment a speech dataset by means of speed. Instead of altering the frequency of an utterance with VTLP or simply applying tempo perturbation, they developed a method that uses speed perturbation which produces a warped time signal and is a combination of VTLP and tempo perturbation. Let α be the warping factor and a given audio signal x(t), then:

$$x'(t) = x(\alpha t)$$

This results in a new length of the audio signal and thus also affects the number of frames for each input signal. Ko et al. (2015) showed, that this method does not only alters the length, and thus the tempo for a given signal, but also shifts the spectrum in a way that is similar to the results of VTLP.

Salamon and Bello (2017) described a method to augment audio datasets by adding background noise to the signal. Therefore, the samples get mixed with different types of noise, for example street and park sounds. This augmentation strategy may improve the results in a lower rate than time or pitch augmentation, but it can be combined with other techniques to obtain further improvements.

Voice conversion can transform an utterance to a different speaker. This way each training example can be used several times. This might be a good approach to augment the speech dataset, but according to Jaitly and Hinton (2013) the number of target speakers is limited, since a lot of speaker individual data is required to develop the models for sufficient voice conversion. Thus, this method is not applicable for low resourced languages.

3.2 Speech Recognition for Amharic

This chapter gives a summary of research in the past that has dealt with speech recognition for Amharic.

Since it is crucial for every automated speech recognition system to have a recorded corpus for training, Abate et al. (2005) developed a read-speech corpus with 20 hours recorded data from 100 different speakers. This corpus is widely used for Amharic speech recognition systems.

Seid and Gambäck (2005) used a hybrid HMM/ANN (Artificial Neural Network) model for speaker independent Amharic speech recognition and showed, that this model is superior to pure HMM models.

Abate and Menzel (2007) used a syllable based HMM-recognizer. This was motivated by the Amharic orthography, which has an almost one-to-one mapping to corresponding

syllabic sounds. According to them this is a promising alternative to tri-phone recognizers.

Gizaw (2008) used a multiple pronunciation model based on a phone based HMM recognizer and concluded, that this approach can improve the performance for Amharic speech recognition.

To improve speech recognition for Amharic, especially for the OOV-problem of this morphologically rich language, Tachbelie et al. (2009) used a morpheme based statistical language model. Their experiments showed, that using morpheme-based language models can bring a slight performance improvement by up to 3% und thus are superior in comparison to word-based language models for Amharic. Since they assumed a closed vocabulary where all words that are tested are trained at least once, it is plausible that on an open vocabulary task where OOVs actually occur the improvement of morpheme-based models is even more effective. With a total word error rate of 7.08% the results look very promising, but the assumption of a closed vocabulary for Amharic as target language is unrealistic.

Melese et al. (2016) used a domain specific speech recognition task for Amharic where they trained the acoustic model on the dataset developed by (Abate et al., 2005) and tested the performance on a self-developed dataset in the tourism domain. They compared morpheme and word-based language models and showed, that morphemes as sub-word units can improve the results compared to word-based language models by 57.39%. The morpheme-based approach showed a WER of 19.6%.

4 Experiments

This chapter will explain the experiments in detail. In section 4.1 the dataset used in the experiments will be described and internal information about it will be provided. Section 4.2 gives an overview about different measures to compare the performance of automated speech recognition systems. Section 4.3 describes the different models used in the experiments to investigate the effects of different data augmentation techniques for different models.

4.1 Data

The dataset used for the experiments was constructed by Abate et al. in 2005. It consists of 10,850 audio-files in .wav format. The dataset is a read-speech corpus, so there is no spontaneous speech in it.

The corpus is already cleaned, in that spelling and grammar errors have been corrected, abbreviations have been written out, foreign words have been eliminated, numbers have been textually written out and concatenated words have been separated.

There are 100 different speakers, 80 of the Addis Ababa dialect who read 100 sentences each and 20 of the other four dialects who read 120 sentences each. For recording, a headset close speaking microphone with noise canceling was used.

Each audio-file contains one spoken sentence in Amharic and has a total length of 103.909 words with 28.615 unique words.

On average each sentence has a length of approximately 9.55 words and each word has a mean-occurrence of approximately 3.63.

Word	Count
ነው	2300
ናቸው	745
ውስጥ	609
የኢትዮጵያ	606
ኢትዮጵያ	580
ላይ	573
ወደ	464
መንግስት	393
ግን	371
ደግሞ	333

Table 2: The ten most frequent words in the dataset with the total number of occurrences

As in all languages, there are words in the dataset that are much more frequent than others. As Table 2 shows, the word 'ነው' (in English equivalent with 'it is') is the most frequent word with a total count of 2,300 occurrences.

Due to the comparably small dataset and the rich morphology of the Amharic language, there are many words in the dataset that are unique and occur only once. The total number of words which occur only once in the whole dataset is 18,755, so from our 28,615 different words and that is 65.5%. These single words represent 18% of the total words in the dataset.

When one of these words occurs in the training-set there is no chance that it will be recognized properly because there are no examples in the training-set.

On the grapheme level, there are 451,766 total graphemes in the dataset with 207 different graphemes. This gives a much better base to cover datapoints, since we have a mean occurrence of ca 2,182 for each grapheme. In contrast to the word level, even the least frequent grapheme occurs several times and each grapheme in the test also occurs in the training set.

Table 3 shows the graphemes of the dataset and how often they occur.

ን	28252	ግ	5043	ሮ	2160	ሲ	1154	ዜ	697	ጭ	463	ጀ	160	ቨ	54
ት	19877	ላ	4795	ሁ	2138	ቻ	1137	ቴ	696	ኘ	429	ጀ	141	ጉ	52
ው	17874	ረ	4485	ሆ	2028	ፐ	1124	ዴ	690	ኪ	419	ኡ	132	ጴ	52
ስ	13957	ብ	4450	ጋ	1972	ጉ	1089	ኾ	688	ፉ	402	ሾ	123	ጢ	51
አ	13556	ድ	4415	ሊ	1940	ዝ	1088	ኔ	657	ጀ	390	ጳ	114	ኼ	50
ያ	13015	ሰ	4413	ቶ	1929	ቢ	1051	ፋ	622	ጓ	361	ጩ	113	ዢ	49
የ	12785	ከ	4213	ካ	1864	ሶ	1003	ዶ	621	ፌ	347	ቪ	111	ፑ	49
ተ	11641	ወ	4150	ቱ	1825	ዚ	986	ጵ	610	ጱ	342	ቸ	111	ጲ	48
በ	11309	ኢ	4033	ፈ	1789	ጀ	982	ፊ	595	ቂ	296	ኰ	109	ኚ	46
ል	10316	ታ	3957	ጣ	1635	ቁ	967	ጮ	590	ኬ	282	ጌ	104	ዣ	46
ለ	10095	ዳ	3812	ፍ	1564	ሱ	960	ለ	582	ኪ	268	ፐ	103	ጰ	46
ር	9527	ክ	3760	ሀ	1558	ኖ	946	ቆ	580	ጦ	255	ኸ	100	ሹ	42
እ	9125	ዮ	3739	ሞ	1531	ኩ	920	ኞ	564	ሹ	254	ቬ	98	ጨ	42
መ	8809	ዋ	3378	ዊ	1516	ቋ	874	ጳ	555	ፎ	251	ጐ	98	ጶ	41
ቸ	8458	ሀ	3032	ሸ	1505	ኒ	855	ሴ	541	ቈ	248	ፔ	94	ጌ	40
ም	8449	ጵ	2850	ዘ	1462	ሌ	854	ጎ	539	ሺ	243	ዥ	86	ጰ	40
ና	8413	ጥ	2791	ሬ	1427	ቹ	853	ጊ	538	ፒ	242	ቄ	84	ቍ	38
ደ	8124	ሪ	2656	ኮ	1412	ቤ	841	ቦ	535	ጆ	237	፪	79	ዦ	38
ነ	7387	ቀ	2614	ኤ	1411	ዱ	814	፩	527	ሂ	236	ቼ	70	ቨ	34
ገ	7155	ጠ	2580	ሎ	1342	ጸ	783	ሄ	524	ቨ	221	ቺ	66	ኩ	33
ማ	6039	ቅ	2568	ኛ	1269	ጡ	763	ጨ	517	ቾ	213	ቫ	65	ዥ	30
ባ	5937	ዲ	2459	ቡ	1266	ሽ	746	ኦ	506	ጮ	195	ጩ	61	ጲ	23
ይ	5651	ዎ	2427	ጮ	1226	ዙ	741	ፖ	506	ዞ	189	ሼ	60	ቄ	20
ሚ	5463	ሩ	2357	ቃ	1210	ኝ	721	ጫ	503	ጤ	188	ቄ	57	ኴ	18
ቸ	5086	ሳ	2355	ዛ	1207	ጅ	718	ሻ	494	ቁ	183	ዤ	55	ጲ	10
ራ	5057	ሉ	2285	ኑ	1202	ቲ	716	ፓ	481	ጪ	164	ጸ	55		

Table 3: The graphemes of the dataset with the total number of occurrences

Compared to large training datasets for popular languages, this dataset is small with 20 hours of recording.

Since the effort of creating a phone-based pronunciation dictionary is very high and requires Amharic linguistic expertise, we took advantage of the syllabic writing system (see section 1.3) and used the Amharic graphemes as base-unit for a syllable based pronunciation dictionary.

For training the language model the Amharic Web Corpus developed by Rychly et al. (2016) was used. It consists of approximately 1,200,000 sentences and was crawled from the web in 2013, 2015 and 2016 by the SpiderLing[1].

Since many sentences in the corpus contain digits, e-mail addresses and other symbols or graphemes that do not belong to the Amharic writing system, these sentences were sorted out and not used to train the language model. The remaining 300,000 sentences were cleaned by removing the punctuation marks.

4.2 Measuring Error Rates

To calculate the errors of the recognition process one compares a reference sentence with the recognized hypothesis sentence. Errors are described in terms of substitution (word of hypothesis and reference differ), deletion (the word is left out in the hypothesis) and

insertions (there is an extra word in the hypothesis) errors. The word error rate *WER* is a combination of them:

$$WER = \frac{N_{sub} + N_{ins} + N_{del}}{N}$$

Where N is the total number of words in the reference sentence, N_{sub} is the total number of substitutions, N_{ins} is the total number of insertions and N_{del} is the total number of deletions.

[1] http://corpus.tools/wiki/SpiderLing (visited on 01/24/2020)

The word error rate is a suitable measure to compare recognition performance within a language but is not always a good choice to compare performance across different languages since it is based on words. (Different languages might have different notions of the concept 'word').

The syllable error rate *SyER* is calculated in a similar way to the word error rate but syllables (in our case the graphemes) are used as base unit instead of words.

Another important error rate in the field of speech recognition is the sentence error rate *SER* which states the relative proportion of incorrectly recognized sentences.

$$SER = 1 - \frac{C}{N}$$

Where *C* is the number of correctly recognized sentences and *N* is the total number of sentences in the test-set.

4.3 Experiment Setup

This thesis compares, how different data augmentation methods affect the performance of speech recognition systems for rare languages. For this purpose, different language models based on different recognition units are developed and the impact on performance when applying data augmentation methods are compared.

As acoustic models the well-known GMM as described in section 2.1.2 which from now on will be written as *GMM*. The other model will be the time delayed neural network as described in section 2.1.3 which from now on will be marked as *TDNN* are used. Due to the lack of linguistic expertise to create a 'real' phoneme-based model, grapheme-based syllables are used as base recognition units for the acoustic models.

To train the acoustic models the dataset developed by Abate et al. (2005) was randomly split into 90% training data and 10% test data.

As language model, the statistical n-gram as described in section 2.3 is used in the experiments.

Two different language models and their influence of data augmentation are compared. For training the language model, the 90% of the sentences from the dataset developed

by Abate et al. (2005) which are also used to train the acoustic model are combined with the 300.000 sentences selected from the dataset that was developed by Rychly et al. (2016). First a standard statistical word-based 3-gram language model is used, which will be called *WORD* from now on. As can be seen in section 4.1 this model suffers from the rich morphology of the Amharic language since the proportion of low frequent and single words is very high. This will result in many out of vocabulary words during the experiments. To avoid the problem of OOVs second model uses a syllable based 4-gram language model. Since the coverage of syllables and sequences of syllables is much higher than for words, we allow a higher n-gram order of 4. This model uses a special silent token <w> to mark the word boundary and is able to detect out of vocabulary words, since it is focused to detect small sub-word units instead of whole words. The syllable-based language model will be marked as *SYL* from now on.

In section 3.1 different augmentation strategies are explained. In our experiments we use four different augmentation approaches. For data augmentation the ffmpeg tool was used, a cross-platform tool for video and audio manipulation[2].

The first strategy is a 3-fold tempo augmentation with manipulation factor of 0.9, 1.0 and 1.1, where 1.0 is the original data, and will be marked as *T*. For this augmentation the following command was used:

```
ffmpeg -i $f -filter:a "atempo=1.1"
ffmpeg -i $f -filter:a "atempo=0.9"
```

The second strategy is a 3-fold pitch augmentation with a manipulation factor of 0.9, 1.0 and 1.1 and will be marked as *P*. For this augmentation the following command was used (the base sample-rate is 8.0K):

```
ffmpeg -i $f -filter:a "asetrate=r=8.8K"
ffmpeg -i $f -filter:a "asetrate=r=7.2K"
```

[2] https://www.ffmpeg.org (visited on 01/24/2020)

The next augmentation approach combines the augmented sets from T and P which results in a 5-fold augmentation and will be marked as T/P.
And finally, the last strategy is a 3-fold speed augmentation with a manipulation factor of 0.9, 1.0 and 1.1 which is basically a combination of tempo and pitch augmentation.
So, in conclusion we can derive 20 different Models:

GMM+WORD:

This model uses a standard tri-phone acoustic model and the 3-gram language model is based on words. No augmentation strategies are applied.

GMM+WORD+T:

Based on the *TRI+WORD* approach but with 3-fold tempo augmentation.

GMM+WORD+P:

Based on the *TRI+WORD* approach but with 3-fold pitch augmentation.

GMM+WORD+S:

Based on the *TRI+WORD* approach but with 3-fold speed augmentation.

GMM+WORD+T/P:

Based on the *TRI+WORD* approach but with both tempo and pitch augmentation which results in a 5-fold augmentation.

GMM+SYL:

This model uses the GMM acoustic model and a syllable-based 4-gram language model with no augmentation on the dataset.

GMM+SYL+T:

Based on the *TRI+SYL* approach but with 3-fold tempo augmentation.

GMM+SYL+P:

Based on the *TRI+SYL* approach but with 3-fold pitch augmentation.

GMM+SYL+S:

Based on the *TRI+SYL* approach but with 3-fold speed augmentation.

GMM+SYL+T/P:

Based on the *TRI+SYL* approach but with combined 5-fold tempo and pitch augmentation.

TDNN+WORD:

This model uses a TDNN acoustic model and the 3-gram language model is word-based with no augmentation strategies.

TDNN+WORD+T:

Based on the *TDNN+WORD* approach but with 3-fold tempo augmentation.

TDNN+WORD+P:

Based on the *TDNN+WORD* approach but with 3-fold pitch augmentation.

TDNN+WORD+S:

Based on the *TDNN+WORD* approach but with 3-fold speed augmentation.

TDNN+WORD+T/P:

Based on the *TDNN+WORD* approach but with combined 5-fold tempo and pitch augmentation.

TDNN+SYL:

This model uses the TDNN acoustic model and a syllable-based 4-gram language model with no augmentation on the dataset.

TDNN+SYL+T:

Based on the *TDNN+SYL* approach but with 3-fold tempo augmentation.

TDNN+SYL+P:

Based on the *TDNN+SYL* approach but with 3-fold pitch augmentation.

TDNN+SYL+S:

Based on the *TDNN+SYL* approach but with 3-fold speed augmentation.

TDNN+SYL+T/P:

Based on the *TDNN+SYL* approach but with combined 5-fold tempo and pitch augmentation.

5 Results

This section presents the results of the experiments. Figure 10 shows the results of all approaches in terms of WER. Generally, in terms of WER the word-based LMs approaches are clearly superior to the *SYL*-based approaches and the TDNNs bring an improvement compared to the GMM-based approaches.

First the results of the GMM based approaches will be shown and after that, the results of the TDNN based approaches will be presented.

The performance of the different models will be compared in terms of WER, SyER and SER.

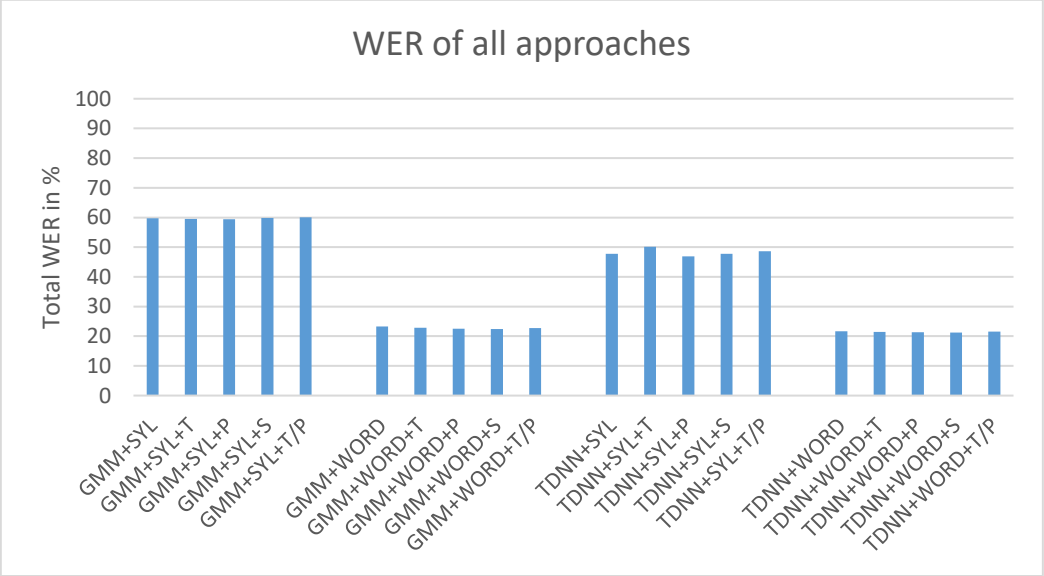

Figure 10: The total WER of all approaches that are compared in this thesis

5.1 GMM-based Results

Table 4 shows the results of the different augmentation strategies based on the Gaussian Mixture Model approaches with words as base-unit for the statistical language model. With the basic *GMM+WORD* we obtained a word error rate of 23.28%, a syllable error rate of 22.92% and a sentence error rate of 46.83%.

Approach:	**WER:**	**SyER:**	**SER:**
GMM+WORD	23.28%	22.92%	46.83%
GMM +WORD+T	22.85%	22.96%	46.37%
GMM +WORD+P	22.54%	22.79%	44.89%
GMM +WORD+S	22.47%	22.64%	44.8%
GMM +WORD+T/P	22.79%	22.9%	45.17%

Table 4: The results in terms of WER, SyER and SER for the *GMM* based approaches and words as base-unit for the language model

All of the data augmentation strategies are leading to an improvement of the results in terms of WER and SER. The speed augmentation (*GMM+WORD+S*) strategy brings the highest improvement with a total WER of 22.47%, which is a relative improvement by 3.48%. In comparison to the *GMM+WORD* strategy, the SER of the *GMM+WORD+S* approach decreases by 2.3% from 46.83% to a total SER of 44.8%.

As can be seen in Figure 11, the relative improvement in terms of WER of the *GMM+WORD+S* approach was the highest, followed by the *P* augmentation. The *GMM+WORD+T* and the *GMM+WORD+T/P* approach had lower impact on the performance where the *GMM+WORD+T/P* showed slightly better improvements in terms of WER.

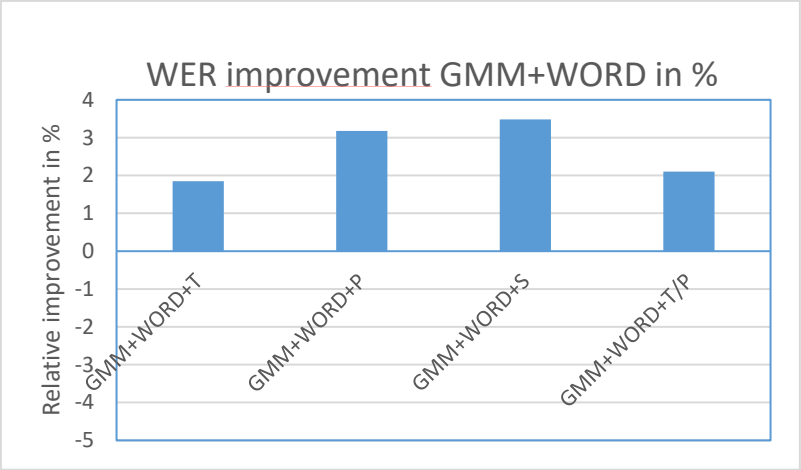

Figure 11: The relative improvement in terms of WER of the *GMM+WORD* based approaches when different augmentation strategies are applied

In Table 5 the results of the *GMM* approach with syllables as base-unit for the language model are presented.

Approach:	**WER:**	**SyER:**	**SER:**
GMM+SYL	59.71%	37.22%	93.38%
GMM+SYL+T	59.52%	37.37%	91.81%
GMM+SYL+P	59.40%	37.71%	93.19%
GMM+SYL+S	59.87%	38.47%	94.02%
GMM+SYL+T/P	60.06%	37.83%	95.12%

Table 5: The results in terms of word error rate (WER), syllable error rate (SyER) and Sentence error rate (SER) for the *GMM* based approaches with syllables as base-unit for the language model

The WER of the basic *GMM+SYL* approach with no augmentation strategy is 59.71%, the SyER 37.22% and the SER 93.38%.

When augmenting the training data with the tempo augmentation strategy (*GMM+SYL+T*), the SyER get worse in that it increases by 0.15%, which is a relative increasement of 0.4%. The *GMM+SYL+P* approach increases the SyER even more by 0.49%, which is a relative increasement of 1.37% in comparison to the basic *GMM+SYL*

approach. In contrast to that, the SER and WER decreases for both approaches. For the *GMM+SYL+T* approach there was a relative improvement of the SER by 1.68% and the absolute SER is 91.81, while with the *GMM+SYL+P* strategy the improvement was smaller and the absolute SER is 93.19, which is a relative improvement of 0.19%.

With the *GMM+SYL+T/P* approach we obtained the worst results in terms of WER. The WER increased to 60.06% and there was also a relative increasement of the SyER by 1.64%, so the absolute SyER is 37.83% and the SER increased to 95.12% which is a relative deterioration of 1.86%.

Both, the *T/P* strategy and the *S* strategy failed to improve the results in terms of WER. The SyER and SER also increased for both approaches. Interesting in this context is that the relative increasement of SyER for the *S* strategy (3.36%) is much higher than for the T/P strategy (1.63%). In contrast to that, the WER and SER of the *S* strategy are slightly better compared to the *T/P* approach.

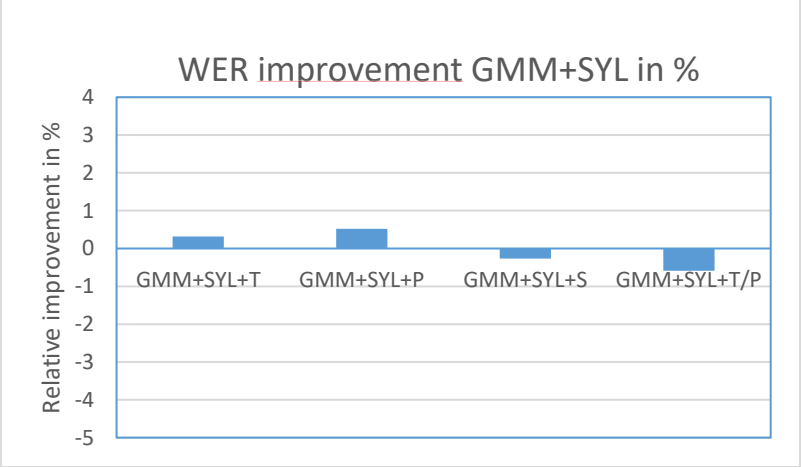

Figure 12: The relative improvement of the WER for the *GMM+SYL* based approaches when different augmentation strategies are applied

When comparing the relative improvements of the word based approaches (Figure 11) and the Syllable based approaches (Figure 12), it is clear to see that the augmentation strategies work much better with the word-based approaches, since the relative improvement of the WER is much higher. For the *SYL* approaches the *P* strategy have

brought the highest improvement and for the *WORD* approaches the S strategy showed the best results.

Figure 13 illustrates the relative improvement and deterioration of the different augmentation strategies in terms of SER. Similar as for the WER, the augmentation strategies where more effective when applied on the *WORD* models. In fact, only the tempo augmentation strategy achieved to improve the results for the *GMM*-based models. For the *WORD* approaches the best improvement in terms of SER was achieved by augmenting the data with the *S* strategy and the tempo augmentation obtains the least improvement.

When using the syllable-based language model, the tempo augmentation results in the largest relative improvement, while the pitch augmentation only gives a minimal improvement in terms of SER and the combined *T/P* and *S* augmentation strategy even worsen the SER.

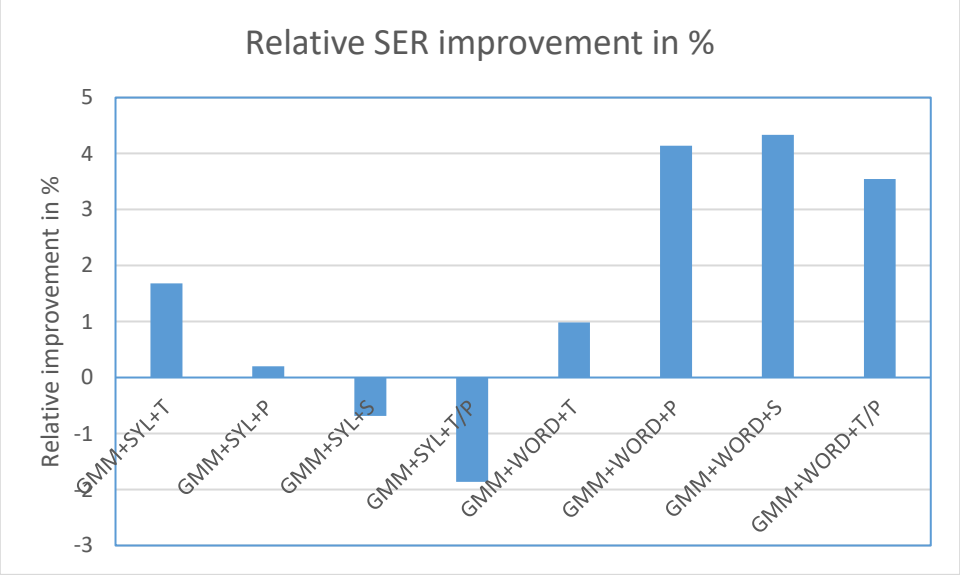

Figure 13: The relative improvement (deterioration) in terms of SER for different augmentation strategies of the *GMM* approaches

5.2 TDNN-based Results

The results of the *TDNN+WORD* and the effect of the different data augmentation strategies are shown in Table 6.

Approach:	**WER:**	**SyER:**	**SER:**
TDNN+WORD	21.67%	22.27%	42.31%
TDNN+WORD+T	21.44%	21.97%	46.37%
TDNN+WORD+P	21.29%	21.91%	41.95%
TDNN+WORD+S	21.22%	21.82%	41.31%
TDNN+WORD+T/P	21.59%	22.1%	43.70%

Table 6: The results in terms of word error rate (WER), syllable error rate (SyER) and Sentence error rate (SER) for the *TDNN* based approaches and words as base-unit for the language model

The basic *TDNN+WORD* approach without any data augmentation technique applied obtained a word error rate of 21.67%, a syllable error rate of 22.27% and a sentence error rate of 42.31%. Similar to the *SYL*-based approaches, all of the data augmentation strategies lead to improved results in terms of WER. The speed *TDNN+WORD+S* strategy scored the best results with a total WER of 21.22%, which is a relative improvement by 2.08%. This approach also scored the best SyER and WER compared to all other approaches used in the experiments.

As you can see in Figure 14, the relative improvement in terms of WER of the *TDNN+WORD+S* approach was the highest, followed by the *P* augmentation, which is similar to the *GMM+WORD* based approaches. In contrast to *GMM+WORD*, the *T/P* showed the lowest improvement for the *TDNN+WORD* approaches.

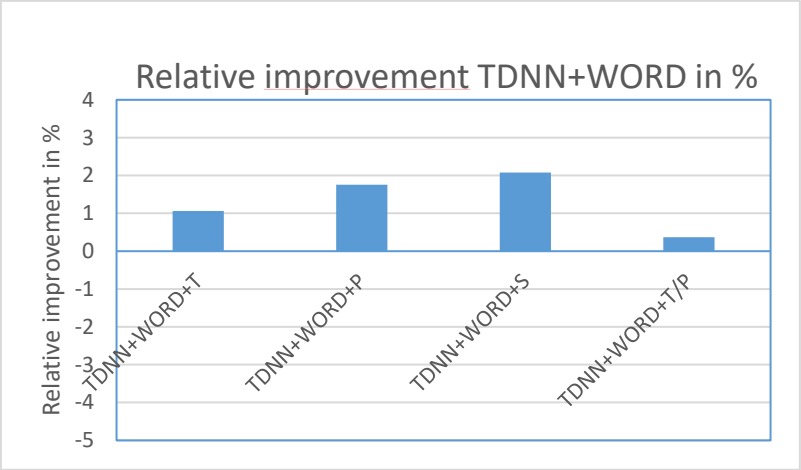

Figure 14: The relative improvement in terms of WER of the *TDNN+WORD* based approaches when different augmentation strategies are applied

Table 7 shows the results of the *TDNN* approaches with syllables as base-unit for the language model.

Approach:	**WER:**	**SyER:**	**SER:**
TDNN+SYL	47.8%	29.92%	83.9%
TDNN+SYL+T	50.14%	31.99%	88.87%
TDNN+SYL+P	46.85%	30.58%	86.38%
TDNN+SYL+S	47.74%	38.47%	94.02%
TDNN+SYL+T/P	48.58%	31.21%	88.68%

Table 7: The results in terms of word error rate (WER), syllable error rate (SyER) and Sentence error rate (SER) for the *TDNN* based approaches with syllables as base-unit for the language model

The basic *TDNN+SYL* approach with no augmentation strategy scored a total WER of 47.8%, a SyER of 29.92% and a SER of 83.9%.

Only the pitch and speed augmentation technique had a positive effect on their performance in terms of WER, where the speed strategies improvements are very low. Similar to the *GMM+SYL* approaches, the SyER increases for all augmentation strategies on the *TDNN+SYL* approaches. Interesting is, that the *S* augmentation showed the lowest performance in terms of SyER but still was able to improve the WER.

The SER of the *TDNN+SYL* approaches get worse when applying any of the augmentation strategies. The highest effect has the *S* strategy, which also obtains the highest SyER increasement, where the SER shows a relative increasement of 12.06%.

With the *TDNN+SYL+T/P* and *TDNN+SYL+T* approaches the WER increases compared to the standard *TDNN+SYL* approach. For the *TDNN+SYL+T/P* strategy, the WER increased to 48.58% which is a relative increasement of 1.63% whereas the WER of the *TDNN+SYL+T* approach increased by 4.89%, so it showed the worst results with a WER of 50.14%.

As Figure 15 shows, the *T* and *T/P* strategies had a negative impact on the WER, the *P* augmentation technique improved the performance by 1.99% and the *S* approach had no great impact on the WER.

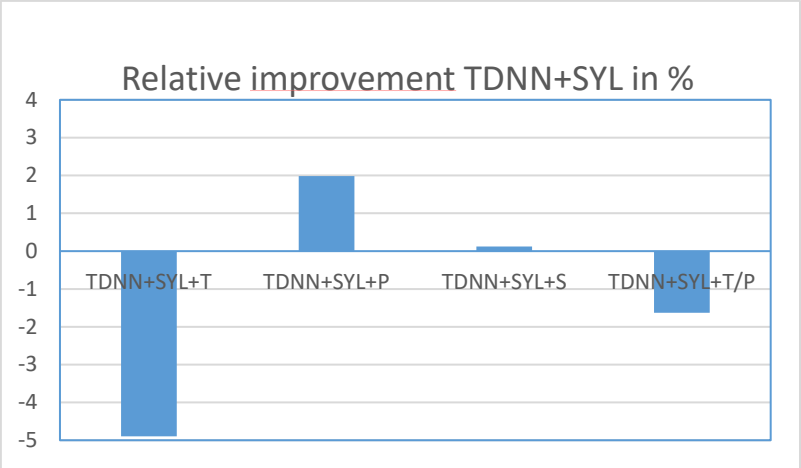

Figure 15: The relative improvement of the WER for the *TDNN+SYL* based approaches when different augmentation strategies are applied

Similar to the *GMM* approaches, the augmentation strategies show better and more reliable improvements when applied on word based LMs instead of syllable based LMs.

Figure 16 illustrates, that the SER for most *TDNN* approaches increases. All of the syllable-based models obtained a worse SER. Of the *WORD* approaches, only the speed and pitch augmentation showed an improving SER.

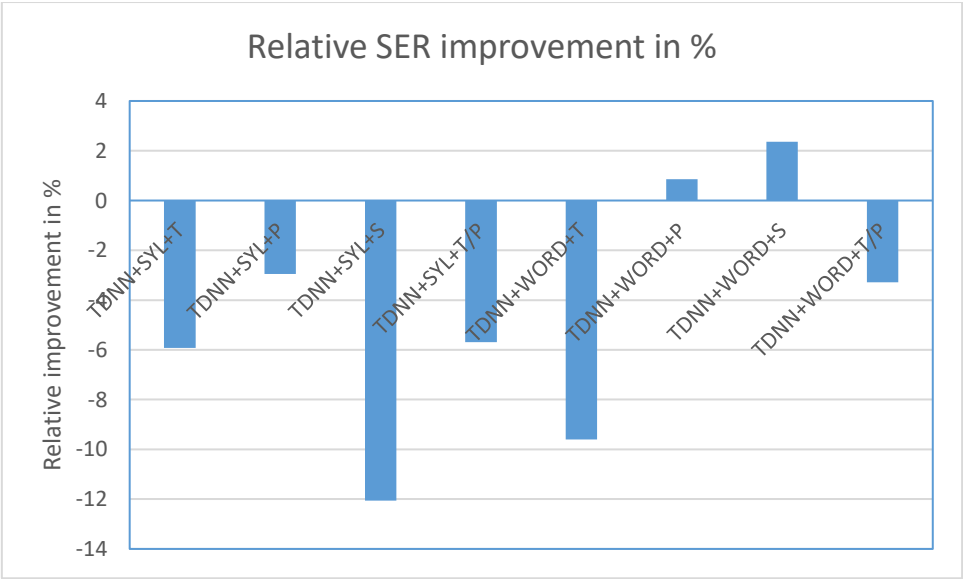

Figure 16: The relative improvement (deterioration) in terms of SER for different augmentation strategies of the *TDNN* approaches

6 Discussion

Using data augmentation techniques to extend the audible dataset is a useful approach to improve the performance of speech recognition systems. This is especially relevant for low resourced languages, where the available data often is strongly limited. This thesis presented three different augmentation approaches, which can be easily applied to any speech recognition dataset.

To answer the key research question, this chapter discusses the sub-questions as mentioned in section 1.2.

1. Is it possible to improve the performance of automated speech recognition systems for low resourced languages with data augmentation techniques?

- **1.1:** Which data augmentation techniques are suitable for low resourced languages?
- **1.2:** Which augmentation strategy works best?

In section 3.1 different augmentation techniques are presented. Most techniques like the speed, tempo and pitch augmentation can be simply applied on single audio files and thus are suitable for low resourced languages. Others like speaker adaption need lots of training data and thus are not applicable for low resourced languages.

For the syllable-based language models, the augmentation strategies show unreliable and both improving and deteriorating effects on the performance. The overall deficient performance of the syllable-based models might be the reason why the data augmentation techniques show these contradictory results.

The experiments showed that for Amharic, out of the compared augmentation techniques (speed, tempo and pitch), the speed augmentation obtains the largest improvement on the word-based language models with a relative improvement of 2.08%.

Since the speed augmentation combines more variation in the augmented files, it is absolutely reasonable that the resulting model learned more robust parameters than the tempo or pitch only augmentation. The *T/P* approach combines the tempo augmented

files and the pitch augmented files and thus learns variations in the frequency and time domain, but the speed augmentation is still superior. This leads to the assumption that, when applying several augmentation techniques it may be better to apply them on top of each other rather than creating new files for each single augmentation method.

2. Is it possible to improve speech recognition for low resourced languages using different acoustic and language models?

2.1: Which acoustic model is best suited for the task of speech recognition for low resourced languages?

2.2: Which language model is best suited for the task of speech recognition for low resourced languages?

The results show, that the TDNN acoustic model is superior to the GMM acoustic model in the experiments with an average relative improvement of 19.24% for the syllable-based LMs and 5.89% for the Word based LMs. As described in section 2.1.2 The advantage of the GMM, that it is highly parallelizable and thus can train on massive amounts of data, cannot be used when it comes to low resourced languages, since there are no large datasets we can train on. The advantage of TDNNs over GMMs is that they can generalize over multiple time delays as explained in section 2.1.3. The drawback that this model needs a lot of computation time for training is manageable for low resourced languages, since the data for training is often very limited.

In the experiments, the word-based language models show much better results compared to the syllable-based language model approaches. The syllable-based approaches obtained word error rates around 50%-60% whereas the WER of the word-based LMs range from 21.22% to 23.28%. For the syllable-based approaches this means that almost half of the words could be recognized correctly by this approach although not a single one of them occurred as a 'word' in the pronunciation dictionary or the language model. The results are probably so poor, because of the fact that this language model can not only recognize every possible word but also every impossible word

(words that do not exist in Amharic) and the proportion of the possible words is much smaller.

3. What is the best combination of augmentation strategy, acoustic model and language model?

Out of the 20 approaches compared in this thesis, the *TDNN+WORD+S* model scored the best WER (21.22%), SER (21.91%) and SyER (41,31). As mentioned before, the word-based language models performed better than the syllable language models, the *TDNN* outperforms the *GMM* based models and the speed augmentation obtained the best improvements in terms of WER. Since the *TDNN+WORD+S* model scores the best results in term of WER, SER and SyER, the different strategies can be combined well to further improve the recognition rate.

4. Since we use Amharic as a low resourced language target, can the concluded model be transferred to other low resourced languages?

Using a TDNN is highly recommended for low resourced languages. Since the datasets for training are often relatively small for low resourced languages, the high computational effort to train a TDNN is acceptable. The generalization over the time improves the performance and outperforms the GMM in the experiments.
The speed augmentation showed the best improve for the Amharic language, but since every language has unique articulation properties, most likely there is no general superior augmentation approach that shows the best improve for every language.

An interesting finding in the results is, that when the performance in terms of WER improves the SyER often worsen. The data-augmentation strategies showed in 12 out of 16 experiments an improvement in terms of WER and only in 7 out of these 12 there was also an improvement of the SyER. Obviously these two measures are related since 0%

WER also means 0% SyER and 100% SyER will result in a 100% WER. Still this observation shows, that a lower WER does not automatically imply a lower SyER.

Graphemes as base acoustic units for the pronunciation dictionary are suitable for Amharic. This has the major advantage, that no linguistic expert knowledge is needed to create the pronunciation dictionary. But this method is not applicable for all languages and only works for Amharic due to the close one-to-one mapping of spoken syllables and written graphemes. Other low resourced languages which share this property could also use graphemes as base for the pronunciation dictionary.

Compared to the results from Tachbelie (2010), who scored a WER of 7.08% and Melese et al. (2016) who scored a WER of 19.1% the results of this thesis seem poorer with a WER of 21.22%. This is probably because this thesis assumes an open vocabulary, while Tachbelie (2010) uses a closed vocabulary with no OOVs and Melese et al. (2016) used a domain specific vocabulary which results in less OOVs compared to an open vocabulary.

7 Future Work

The experiments showed that training a model on an augmented dataset can improve the performance of speech recognition systems for Amharic. In our experiments different augmentation techniques were compared and the speed augmentation showed the best results. The augmentation techniques used in this thesis rely on the time and frequency dimension. It is also plausible to augment the dataset by changing the volume, add noise and echo or emulate different spatial environments. Comparing the effect of other combinations of augmentation strategies may bring more insight to how to improve the models for low resourced languages. Another interesting field of research would be if the augmentation techniques show similar improvements for other low resourced languages than Amharic.

Using a grapheme-based language model to detect out of vocabulary words showed a strongly decreasing performance. Researches in the past have shown, that using morphemes instead of words can improve the performance (Tachbelie et al., 2009; Melese et al., 2016). Since morphemes, like graphemes, are sub word units, it is possible to detect out of vocabulary words with morpheme-based language models. A TDNN-based Speech recognition system that uses a morpheme-based language model and a speed augmented dataset will probably get better results than the proposed *TDNN+WORD+S* approach.

8 References

Abate, S. T. and W. Menzel (2007). "Automatic speech recognition for an under-resourced language-amharic". In: *Eighth Annual Conference of the International Speech Communication Association*.

Abate, S. T., W. Menzel and B. Tafila (2005). "An Amharic speech corpus for large vocabulary continuous speech recognition". In: *Ninth European Conference on Speech Communication and Technology*.

Besacier, L., E. Barnard, A. Karpov and T. Schultz (2014). "Automatic speech recognition for under-resourced languages: A survey" *Speech Communication* 56, 85–100.

Fukada, T. and Y. Sagisaka (1997). "Automatic generation of a pronunciation dictionary based on a pronunciation network". In: *Fifth European Conference on Speech Communication and Technology*.

Gales, M., S. Young and others (2008). "The application of hidden Markov models in speech recognition" *Foundations and Trends\textregistered in Signal Processing* 1 (3), 195–304.

Gizaw, S. (2008). "Multiple pronunciation model for Amharic speech recognition system". In: *Spoken Languages Technologies for Under-Resourced Languages*.

Isern, N. and J. Fort (2014). "Language extinction and linguistic fronts" *Journal of the Royal Society Interface* 11 (94), 20140028.

Jaitly, N. and G. E. Hinton (2013). "Vocal tract length perturbation (VTLP) improves speech recognition". In: *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*.

Jurafsky, D. and J. H. Martin (2014). *Speech and language processing:* Pearson London.

Killer, M., S. Stuker and T. Schultz (2003). "Grapheme based speech recognition". In: *Eighth European Conference on Speech Communication and Technology*.

Ko, T., V. Peddinti, D. Povey and S. Khudanpur (2015). "Audio augmentation for speech recognition". In: *Sixteenth Annual Conference of the International Speech Communication Association*.

Ko, T., V. Peddinti, D. Povey, M. L. Seltzer and S. Khudanpur (2017). "A study on data augmentation of reverberant speech for robust speech recognition". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220–5224.

Kramer, R. (2009). *Definite markers, phi-features, and agreement: A morphosyntactic investigation of the Amharic DP:* Citeseer.

LeCun, Y., L. Bottou, Y. Bengio, P. Haffner and others (1998). "Gradient-based learning applied to document recognition" *Proceedings of the IEEE* 86 (11), 2278–2324.

Leslau, W. (2000). *Introductory grammar of Amharic:* Otto Harrassowitz Verlag.

Melese, M., L. Besacier and M. Meshesha (2016). "Amharic speech recognition for speech translation". In:

Peddinti, V., D. Povey and S. Khudanpur (2015). "A time delay neural network architecture for efficient modeling of long temporal contexts". In: *Sixteenth Annual Conference of the International Speech Communication Association*.

Ragni, A., K. M. Knill, S. P. Rath and M. J. Gales (2014). "Data augmentation for low resource languages".

Reynolds, D. A. (2009). "Gaussian Mixture Models" *Encyclopedia of biometrics* 741.

Rychly et al. (2016). "Annotated amharic corpora". In: *International Conference on Text, Speech, and Dialogue*, pp. 295–302.

Salamon, J. and J. P. Bello (2017). "Deep convolutional neural networks and data augmentation for environmental sound classification" *IEEE Signal Processing Letters* 24 (3), 279–283.

Sarah Adam (2019). *The Most Spoken Languages In Ethiopia 2019*. URL: https://ethiopiangazette.com/most-spoken-languages-in-ethiopia-2019.

Seid, H. and B. Gambäck (2005). "A speaker independent continuous speech recognizer for Amharic".

Singh, R., B. Raj and R. M. Stern (2002). "Automatic generation of subword units for speech recognition systems" *IEEE Transactions on Speech and Audio Processing* 10 (2), 89–99.

Stüker, S., T. Schultz and A. Waibel. "Automatic Generation of Pronunciation Dictionaries".

Tachbelie, M. Y. (2010). "Morphology-based language modeling for Amharic".

Tachbelie, M. Y., S. T. Abate and W. Menzel (2009). "Morpheme-based language modeling for amharic speech recognition". In: *The 4th Language and Technology Conference*.

Thangarajan, R., Am Natarajan and M. Selvam (2008). "Word and triphone based approaches in continuous speech recognition for Tamil language" *WSEAS transactions on signal processing* 4 (3), 76–86.

Vries, N. J. de, M. H. Davel, J. Badenhorst, W. D. Basson, F. de Wet, E. Barnard and A. de Waal (2014). "A smartphone-based ASR data collection tool for under-resourced languages" *Speech Communication* 56, 119–131.

Waibel, A., T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang (1989). "Phoneme recognition using time-delay neural networks" *IEEE transactions on acoustics, speech, and signal processing* 37 (3), 328–339.

Young, S. J., J. J. Odell and P. C. Woodland (1994). "Tree-based state tying for high accuracy acoustic modelling". In: *Proceedings of the workshop on Human Language Technology*, pp. 307–312.

Zhang, Y. and J. R. Glass (2009). "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams". In: *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pp. 398–403.

Zhu, X. and R. Rosenfeld (2001). “Improving trigram language modeling with the world wide web”. In: *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, pp. 533–536.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Studiengang M. Sc. Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, 31.01.2020

_____

Unterschrift