TURKISH LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION BY USING LIMITED AUDIO CORPUS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DERYA SUSMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

FEBRUARY 2012

Approval of the thesis:

TURKISH LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION BY USING LIMITED AUDIO CORPUS

submitted by **DERYA SUSMAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _______________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _______________
Head of Department, **Computer Engineering**

Prof. Dr. Adnan Yazıcı
Supervisor, **Computer Engineering Dept., METU** _______________

Dr. Selçuk Köprü
Co-Supervisor,**Teknoloji Yazılımevi** _______________

Examining Committee Members:

Assist. Prof. Dr. Sinan Kalkan
Computer Eng. Dept., METU _______________

Prof. Dr. Adnan Yazıcı
Computer Eng. Dept., METU _______________

Dr. Selçuk Köprü
Teknoloji Yazılımevi _______________

Assist. Prof. Dr. Tolga Çiloğlu
Electrical and Electronics Eng. Dept., METU _______________

Assist. Prof. Dr. Mustafa Sert
Computer Eng. Dept., Başkent Uni. _______________

Date: 24.02.2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Derya Susman

Signature :

ABSTRACT

TURKISH LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION BY USING LIMITED AUDIO CORPUS

Susman, Derya
M.Sc., Department of Computer Engineering
Supervisor: Prof. Dr. Adnan Yazıcı
Co-Supervisor: Dr. Selçuk Köprü

February 2012, 65 Pages

Speech recognition in Turkish Language is a challenging problem in several perspectives. Most of the challenges are related to the morphological structure of the language. Since Turkish is an agglutinative language, it is possible to generate many words from a single stem by using suffixes. This characteristic of the language increases the out-of-vocabulary (OOV) words, which degrade the performance of a speech recognizer dramatically. Also, Turkish language allows words to be ordered in a free manner, which makes it difficult to generate robust language models.

In this thesis, the existing models and approaches which address the problem of Turkish LVCSR (Large Vocabulary Continuous Speech Recognition) are explored. Different recognition units (words, morphs, stem and endings) are used in generating the n-gram language models. 3-gram and 4-gram language models are generated with respect to the recognition unit.

Since the solution domain of speech recognition is involved with machine learning, the performance of the recognizer depends on the sufficiency of the audio data used in acoustic model training. However, it is difficult to obtain rich audio corpora for the Turkish language. In this thesis, existing approaches are used to solve the problem of Turkish LVCSR by using a limited audio corpus. We also proposed several data selection approaches in order to improve the robustness of the acoustic model.

Keywords: large vocabulary continuous speech recognition, agglutinative, hidden markov model, n-gram language model, low-resourced languages

ÖZ

KISITLI SES KÜLLİYATI İLE TÜRKÇE GENİŞ DAĞARCIKLI SÜREKLİ KONUŞMA TANIMA

Susman, Derya
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Adnan Yazıcı
Ortak Tez Yöneticisi: Dr. Selçuk Köprü

Şubat 2012, 65 sayfa

Türkçe konuşma tanıma çeşitli açılardan zorlu bir problemdir. Zorlukların birçoğu dilin morfolojik yapısından ileri gelmektedir. Türkçe sondan eklemeli bir dil olduğundan, tek bir gövdeden ekler kullanarak birçok yeni kelime oluşturulabilir. Dilin bu özelliği dağarcık dışı kelime sayısını arttırır; bu kelimeler de konuşma tanıyıcısının performansını önemli ölçüde düşürürler. Ayrıca, Türkçe dilinin serbest kelime dizilimine izin vermesi, sağlam dil modelleri oluşturmayı da güçleştirir.

Bu tezde, Türkçe GDSKT (Geniş Dağarcıklı Sürekli Konuşma Tanıma) problemini çözmeyi hedefleyen yaklaşımlar ve modeller araştırılmıştır. N-gram dil modellerini oluştururken değişik tanıma birimleri (kelimeler, morflar, kökler ve ekler) kullanılmıştır. Tanıma birimine uygun olarak, 3-gram ve 4-gram dil modelleri üretilmiştir.

Konuşma tanımanın çözüm alanı makine öğrenmesi ile ilişkili olduğundan, tanıyıcının performansı akustik modeli oluşturmada kullanılan eğitim verisinin yeterliliğine bağlıdır. Ancak, Türkçe dili için zengin bir ses külliyatı elde etmek zordur. Bu tezde, kısıtlı bir ses külliyatı ile var olan yaklaşımlar kullanılarak Türkçe GDSKT problemi çözülmeye çalışılmıştır. Ayrıca, akustik modelin başarımını arttırmak için çeşitli veri seçme yaklaşımları önerilmiştir.

Anahtar Kelimeler: geniş dağarcıklı sürekli konuşma tanıma, sondan eklemeli, saklı markov modeli, n-gram dil modeli, kısıtlı kaynaklı diller

To my family

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude and appreciation to Prof. Adnan Yazıcı and Dr. Selçuk Köprü for their sincere encouragement and profound guidance which helped me a lot during the course of this research.

I would like to thank Tolga Çiloğlu for providing the audio corpus utilized in this thesis.

I would like to thank Mehmet Tatlıcıoğlu and Berker Batur for helping me during the early stages of this thesis.

I would like to thank Haşim Sak for making available a large text corpus and a morphological parser in his web-site and his endurance for answering my questions.

I would also like to thank Ebru Arısoy and Özgül Salor for helping me with their valuable opinions.

Mehtap Kaya, Osman Nuri Osmanlı, Sermet Reha Yavuz, Ersin Er and all my friends who have been with me during the course of this thesis; I am very grateful to you all. Thanks for bearing my grumblings and being with me through the hard times as well as the joyful times.

Finally, I can not thank my family enough for believing in me, not only for this thesis, but for every single step that I take in life. This thesis is dedicated to them.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

LVCSR	Large Vocabulary Continuous Speech Recognition
OOV	Out Of Vocabulary
ASR	Automatic Speech Recognition
WER	Word Error Rate
CWR	Correct Word Rate
WAR	Word Accuracy Rate
HMM	Hidden Markov Model
RTF	Real Time Factor
LM	Language Model
MFCC	Mel-frequency cepstral coefficients
HTK	HMM Toolkit
LDC	Linguistic Data Consortium

CHAPTER 1

INTRODUCTION

In this thesis, we aim to explore the existing approaches and models in order to solve the problem of Turkish LVCSR by using a limited audio corpus. The main challenges in Turkish LVCSR are caused by the productive morphological structure of the language. Also, the availability of sufficient audio corpus is a problem. We aim to show how existing approaches for Turkish perform when limited audio training data is available.

Even without the challenges introduced by the morphological structure of the target language, speech recognition is a complex task from several perspectives. The way a word is uttered differs from person to person. Even, the way a word is uttered by the same person also differs from time to time, depending on her mood and the acoustic conditions of the environment. The perception of the listener is also a factor.

Speech recognition is utilized in various areas. It is an important tool in human and computer interaction where the human is too busy to command the computer by using the keyboard. For example, a speech recognizer in telephony domain can be used to route calls according to the names uttered. In medical areas, dictation systems are used by doctors to prepare medical prescriptions for patients. Different algorithms and techniques are required for different application areas since some of these recognizers are speaker-dependent and some of them are not.

Broadly speaking, a speech recognizer treats the input signal as a sequence of phonetic symbols. The recognizer conducts a search among many candidates to find the word that matches the input signal best.

Hidden Markov Models (HMM) are the most commonly used technique in speech recognition. An HMM is used to represent the speech signal in a statistical approach. Being a state machine at its core, an HMM consists of hidden states which are invisible and observable outputs which are visible. Hidden Markov Models became popular in speech recognition during 1980s.

Ideally, an LVCSR system is expected to recognize every word uttered by the speaker in a natural manner. Although great progress has been achieved in speech recognition since late 90s, it can still be referred as a developing field.

LVCSR systems aim to recognize words in a language as they are uttered by the speaker in a natural and continuous manner. Hence, an LVCSR system is different from an isolated-word speed recognition system in the sense that there is not an intended pause between the words being uttered. Also, building an LVCSR system requires large amounts of text and audio data in order to model the acoustic and lingual properties of the target language.

The sufficiency of the training data is very relevant to the performance of the recognizer because speech recognition heavily utilizes machine learning. If a word is not contained by the audio corpus which is used to train an LVCSR system, there is no possibility of the word to be recognized correctly and the word is deemed to be an out-of-vocabulary (OOV) word. As it is shown in [1], an OOV word introduces 1.5 recognition errors on average. Hence, OOV words are important factors which increase WER (Word Error Rate) values.

A speech recognizer mainly utilizes two models in the recognition process: the acoustic model and the language model. The acoustic model provides the probability of an acoustic observation sequence given a word. The language model gives us the probability of a sequence of words. The coordination of these models is very crucial in an ASR system.

Turkish is an agglutinative language, which enables the generation of many words from a single stem. The productive morphology of Turkish language increases the number of OOV words in a speech recognition process. Language models which use sub-units such as morphs and stem/endings are preferable for Turkish LVCSR [2], [3], [4] and [5], since processing words as they are do not address the problem of high OOV word rates. On the other hand, language models built by using sub-units introduce the problem of generating invalid sequences of sub-units [2].

N-grams are the most widely used approach in building language models for LVCSR systems. Since there is a trade-off between n-gram order and robustness of n-gram parameters, 3-grams are the most widely used n-gram order in ASR (Automatic Speech Recognition) systems [2]. However, for sub-word units, 4-gram models are also preferred.

The amount of the training data utilized in learning based systems is significant in terms of model robustness and computational cost. Data selection can be applied to select a subset which represents the training data best. The selection of a best representative subset is significant if training the system by using the entire training data is not computationally feasible. If the sufficiency of the available training corpus is low, data selection can be adopted to select the more useful data segments and generate artificial samples of the selected data in the training corpus. As pointed out earlier, it is difficult to obtain rich audio corpora for the Turkish language. In order to address the insufficiency of the audio corpus, we employed several data selection approaches. These approaches aim to increase the recognition performance by analyzing and reorganizing the audio training data. We obtained a 1.77% decrease in the word error rate (WER) by reorganizing the audio corpus utilized in this research.

1.1 Statement of the Problem

Since Turkish is a highly productive, agglutinative language, there are more challenges to overcome in the context of speech recognition with respect to other languages. The highly productive nature of the language enables the generation of many distinct words by using a single stem and various suffixes. This property of the language imposes the problem of acquiring large audio training corpora, since the solution domain of speech recognition is related to machine learning.

Previous researches carried out on Turkish LVCSR mainly focus on experimenting with different language modeling approaches. Since it is easy to have many distinct words from a single stem and difficult to obtain a very large audio corpus, language model alternatives to the word-based model were investigated. [4] is one of the first studies which proposes to generate the language model for Turkish by using units smaller than a word.

In [6], stems, endings and morphemes were used to generate a hybrid language model. It is reported that the word based language model outperformed the proposed hybrid language model. In [7], syllable-based language model was compared with stem ending based and word based models. In [4], stem based language modeling was compared with stem ending based language modeling. In that study, recognition results with the stem based model are reported to be better than the results obtained by using the stem ending based model. In [2], morph based language modeling gave the best recognition results. Different amounts of audio data were utilized in these researches.

In our thesis, we show how different language modeling approaches perform when limited amount of audio training data is available. In our research, we experimented with word based, stem based, stem ending based and morph based language models. We also experimented with a hybrid model which is based on word based and stem ending based language modeling. In our experiments conducted with limited audio training data, we obtained the best recognition results by using a word based language model.

1.2 State-of-the-art in Automatic Speech Recognition

The state-of-the-art LVCSR systems are comprised of four main components: front-end processor, acoustic model, language model and decoder [8]. The quantitative representation of the speech signal is obtained by front-end processing. MFCC (Mel-frequency cepstral coefficients) is a commonly used method for acoustic feature extraction in speech recognition domain. The acoustic model is generated by using Hidden Markov Models. N-grams are heavily utilized in generating language models [8]. The decoder component, which encapsulates the Viterbi algorithm at its core, is used to find the best match for the speech input by incorporating the acoustic and the language models.

In state-of-the-art, the ASR systems which target agglutinative and highly productive languages utilize sub-word based language models. Sub-word recognition units can be obtained by using morphological or statistical approaches. Stem-endings, morphemes, syllables and morphs are used as sub-word recognition units [2], [4], [5] and [7].

Recently, speech recognition applications have become more ubiquitous since the numbers of resources available through the Internet and computation power have increased. For instance, one can build an ASR system for the English language by using several freely available speech data and tools. Outside the scope of this thesis, we built an LVCSR system with a moderate recognition rate for the English language by utilizing speech data which is publicly available on the Internet.

Google offers voice-based web search in which the users supply queries in their native languages [37]. Mercedes-Benz conducted speech recognition researches in order to assist drivers in controlling their environmental and navigational systems

[38]. Speech recognition is also utilized by video search companies, such as Blinkx [39], in order to recognize specific words in videos.

With the recent advance of the mobile technologies, speech recognition applications have become more accessible and available. Apple Siri (Speech Interpretation and Recognition Interface) enables users to issue voice-based commands in their mobile phones [40]. Google provides speech recognition services for Android platforms with high recognition rates. The high performance of Google's speech recognition can be related to the large amounts of available voice data retrieved by various speech related services offered by Google.

In recent years, various achievements were obtained in the domain of Turkish LVCSR systems. In [2], a word error rate of 22.9% was obtained by using morphs, which are statistically derived units, as recognition units. In [2], significantly large amounts of acoustic (~194 hours) data were utilized in generating the acoustic models. However, a publicly available, rich Turkish audio corpus which can be used for speech recognition purposes is still not available.

In Turkish speech recognition industry, the Dikte ASR system [41] is available for purposes like writing reports and converting court records to texts. Dikte provides speech recognition capabilities for various domains including medicine and law. The program also provides features which aim to aid the visually-impaired users.

1.3 Organization of the Thesis

In Chapter 2, we explain the fundamental concepts of speech recognition, like acoustic modeling, language modeling and decoding. In this chapter, we also give a brief explanation on Hidden Markov Models and performance measures used in this research.

Chapter 3 narrows down the speech recognition concept to Turkish speech recognition. In this chapter, we explain the morphological structure of Turkish briefly and outline the challenges of speech recognition in Turkish. We also give a literature survey about the existing studies on Turkish LVCSR.

In Chapter 4, we give details about the acoustic and textual data as well as the tools utilized in this research.

In Chapter 5, we explain the details of the approaches utilized in this thesis.

In Chapter 6, the experimental results we obtained in this research are explained and compared with existing studies.

Finally, Chapter 7 concludes this thesis with a summary of our experiments.

CHAPTER 2

SPEECH RECOGNITION

At its core, speech recognition can be considered as a search problem. A speech recognizer accepts a string of symbols as input and tries to return the best matching string of symbols by analyzing all possible strings. The recognizer chooses the best match by considering the conditional probabilities of all possible strings. The output of the recognizer can be used differently with respect to the problem: It can be converted to text or it can be input to another command-based application.

Although the aim of a speech recognizer is to generate the corresponding transcription of speech input, several aspects must be considered with respect to the context in which the speech recognizer will operate. Vocabulary size, speaking manner and environmental conditions are key points to be considered in the design process of a speech recognizer. The vocabulary size which is required by an LVCSR system and a simple digit recognizer is significantly different. A speech recognizer which is designed to operate on a few commands has a pre-determined vocabulary and speaking manner. However, an LVCSR system is supposed to handle natural and spontaneous speaking styles. A speech recognizer which is designed for outdoor use must be adapted to noisy conditions better than a speech recognizer which is designed to operate indoors.

In order to recognize an utterance, a speech recognizer first needs to convert the speech signal into a quantitative format. Next, the speech recognizer conducts a search to find the best matching pattern for the speech input. Modern ASR (Automatic Speech Recognition) systems rely on statistical principles which are based on training strategies. The recognition task is handled by the resultant probabilistic model which is formed by training the system by using audio and textual data.

Next, we'll explain the components of an ASR system.

2.1 Components of an ASR System

The input source of an ASR system is a speaker. The words uttered by the speaker arrive at the front-end processing module of the ASR system over a transmission channel. The speaker, the transmission channel and the front-end processing module are referred as the "noisy channel" [8]. The front-end processor converts the raw speech waveform into acoustic observations. Then, the decoder component of the ASR system tries to choose the sequence of words which best matches the speech input. The decoder utilizes the acoustic and language models during the decoding process. We can outline the main components of an ASR system as below [8]:

- Front-end processor
- Acoustic model
- Language model
- Decoder

The front-end processor converts the speech waveform into multiple frames of 10 to 25 milliseconds. These frames are then transformed into spectral features which determine the energy levels contained by the acoustic signal. In order to recognize individual phones, the acoustic model considers the probabilities of each phoneme (the smallest meaningful unit of sound) in the language. By using these probabilities, the acoustic model determines if a sequence of phones correspond to a word in the language. After phone recognition, the decoder component attempts to find the most probable sequence of words by using a pronunciation dictionary and a language model. The language model enables the decoder to search among sentences which are likely to match the speech input. The probability of a sentence is determined by considering the probabilities of the words which form the sentence. These probabilities are provided by the language model which attempts to model the linguistic properties of the language [8].

The problem to be solved by a speech recognition system can be stated as follows [8]:

“Given an acoustic input O and a language L, what is the most likely sentence for acoustic input O among all sentences in language L?”

The acoustic input is treated as a sequence of observations. Each observation is 10 to 25 ms long. These observations represent the time intervals contained by the acoustic input. So, the acoustic input can be represented as a sequence of individual symbols as shown below:

$$O = o_{1,}o_2,o_3,\ldots,o_n$$

Similarly, a sentence is composed of a sequence of words. A speech recognizer tries to find the most likely word out of all words in the language. This problem can be formulated as:

$$\hat{W} = \underset{W\in L}{\text{argmax}}\ P(W|O) \qquad (2.1)$$

By using the Bayes’ rule, Equation 2.1 can be decomposed as:

$$\hat{W} = \underset{W\in L}{\text{argmax}}\ P(O|W)\ P(W)\ /\ P(O) \qquad (2.2)$$

Here, P(O|W) is the acoustic probability of the observation O to represent the word W. P(W) is the probability of the word W, which is estimated by using the language model. P(O) is the probability of observing the acoustic input. Since P(O) holds the same value for every sentence in the language, this probability can be omitted from the formula. So, the formula can be simplified as:

$$\hat{W} = \underset{W \in L}{\operatorname{argmax}} \, P(O|W)\, P(W) \qquad (2.3)$$

Equation 2.3 summarizes the problem of speech recognition. The most likely sequence of words for an acoustic input can be determined by calculating the product of two possibilities for every sentence in the language and selecting the sentence with the greatest product value [8].

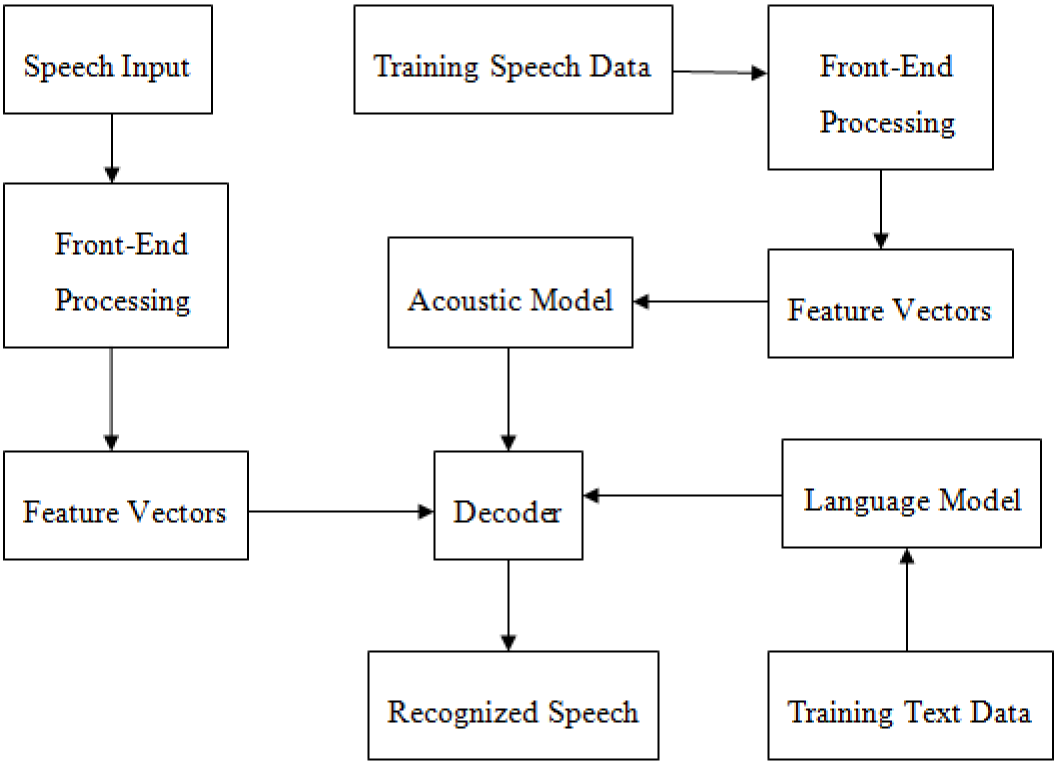


Figure 2.1 - Components of an LVCSR system

In Figure 2.1, an overview of an LVCSR system is given. Initially, an LVCSR system converts a speech input into feature vectors. The acoustic model, which contains phone likelihoods, is obtained by training the system with speech data. The language model, which consists of n-grams, is obtained by training the system with text data. The decoder processes the parameterized speech data by using the acoustic and language models and finally produces the recognized speech.

Next, we will explain the components of an LVSCR system in detail.

2.2 Front End Processing

The speech signal must be converted into a format which can be processed by a computer. This is obtained by acoustic processing, which is referred as feature extraction. In order to represent a slice of a speech signal, a vector of numbers is required. A feature consists of a vector of numbers. By assigning a feature for each and every slice of a speech data, we obtain a quantitative representation for the entire speech signal which can be processed by a computer.

In order to obtain slices of a speech signal, very short time frames (10ms to 25ms) need to be captured. These frames must be small enough so that they can be treated as stationary (i.e., not changing). By using these frames, features are obtained.

Features can be represented by using different techniques such as, linear prediction coefficients (LPC), mel-frequency cepstrum coefficients (MFCC), LP based cepstra or spectrum coefficients [4].

Since it is a widely used format in speech recognition, we utilized the MFCC format in this thesis. More information on acoustic processing of speech can be found in [8].

2.3 Acoustic Model

The aim of the acoustic model component of an ASR system is to compute the probability value, P(O|W), in Equation 2.3. P(O|W) is the acoustic probability of the observation O to represent the word W. Since it is not possible to model each and every word in a language, smaller units like phones and triphones are modeled. By using the models for smaller units, word models are obtained. Hidden Markov Models (HMMs) are the most commonly used models for acoustic modeling in the state-of-the-art speech recognition systems. In the next section, we will explain the essential concepts of an HMM and its use in acoustic modeling. More information about the introduction of HMMs into speech recognition can be found in [9].

2.3.1 Hidden Markov Models

A Markov chain is a weighted automaton. The automaton goes through different states with respect to the input sequence. Each state transition is associated with a probability value.

Hidden Markov Models are special cases of Markov chains. An HMM is treated as a Markov model with unobservable states. An HMM contains observable outputs, which depend on the unobservable states. Every state of the HMM has a probability distribution over the possible observations.

A Markov model consists of:

- A set of states
- A set of transition probabilities
- A start state
- An end state
- A set of observation likelihoods

An HMM introduces two more properties into the standard Markov model:

- A separate set of observations
- The observation likelihoods can take values in [0.0, 1.0] range.

In the context of speech recognition, the states of an HMM correspond to phone symbols. The spectral features obtained by front end processing (see section 2.2) are treated as observable outputs. Each phone symbol corresponds to a number of spectral features with different probabilities.

The pronunciation of a single phone is affected by neighbour phones. To model the pronunciation of single phone correctly, the left and right contexts of the phoneme are also considered. This is achieved by building left-to-right, 3-state HMMs. The triphone acoustic model enables us to consider a monophone in a larger context. Since the recognizer tries to match a group of phonemes instead of a single phoneme, triphones increase the recognition accuracy of an ASR system. Since word models are obtained by concatenating the phone models, modeling the accurate pronunciation of a phone is crucial.

Figure 2.2 depicts the triphone model for the Turkish word "bir". The triphones for the word can be given as: "ε – b + i", "b – i + r" and "i – r + ε"

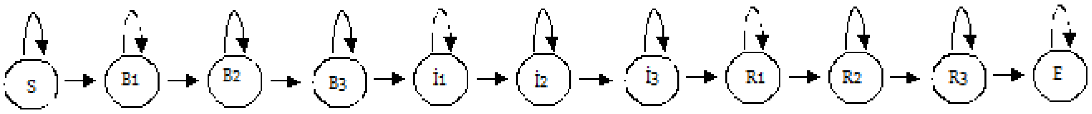


Figure 2.2 - Triphone HMM for the Turkish word "bir"

Being a statistical model, an HMM needs to be trained on a sufficient set of acoustic data in order to learn the acoustic parameters of the model. To find out the most probable phone sequence for a given acoustic observation, the Viterbi algorithm is applied over the trained acoustic model.

Developed by Andrew Viterbi in 1967, the Viterbi algorithm is commonly used in ASR systems to solve the problem of decoding, namely, finding out the most likely state transition path corresponding to an observation sequence. The Viterbi algorithm is a more efficient variation of the Forward algorithm. In the Forward algorithm, each word is considered separately. However, the Viterbi algorithm considers all the words in parallel while finding out the most likely path. More about information about the Viterbi algorithm can be found in [8, 9].

2.4 Language modeling

In speech recognition, predicting the next word which will follow a certain word is an important task. In order to obtain estimations about the word distributions in a particular language, a statistical model is required. This statistical model is referred as the language model. Language models are generated by using significant amounts of training text data. Therefore, the accuracy of a language model is closely dependent on the training text data.

Language modeling aims to compute the P(W) probability in Equation 2.3. Considering the individual frequency of a word is not sufficient to calculate the probability of a word [8]. For instance, if we consider the English words “the” and “voice”, the word “the” is expected to be much more frequent than the word “voice”. However, if we consider a sentence such as “He has a rather soft …” the word “voice” fits in better, no matter how frequent the other word may be.

In estimating the probability of a word, considering the previously uttered words is important because such a history reduces the set of possible words which will follow the previous ones. For instance, if we consider the sentence, “The sun is shining bright”, the probability of the word “bright” can be stated as:

$$P(\text{bright}|\text{The sun is shining})$$

However, calculating such a word probability for long sentences is problematic [8]. This problem can be addressed by keeping history set short. For instance, the bigram model suggests considering only the single previous word. With respect to the bigram model, the probability of the word “bright” can be given as:

$$P(\text{bright}|\text{shining})$$

If the history size is set to three, then the model is a trigram (3-gram) model. The trigram model suggests considering the previous two words. If the history size is set to zero, then the history context is avoided.

An n-gram language model predicts the next word by considering the n previous words. The language models of the state-of-the-art ASR system are composed of n-grams. N-gram models are generated by using training text data. It is important to note that the size and the domain of the training text corpus are crucial factors which affect the accuracy of an n-gram model.

Since an n-gram model is obtained by using a text corpus, a word which is missing in the training text corpus will have zero probability. In order to overcome this problem, smoothing must be applied. During the smoothing process, low probability (including zero probability) n-grams are reevaluated and assigned non-zero probabilities.

In order to have non-zero word probabilities, further actions can be taken. There are two strategies for n-grams: back-off and deleted interpolation. In back-off strategy, the lower n-grams are not taken into account if the n-gram count is greater than zero. For instance, if a trigram model is used, the back-off strategy does not interpolate the bigram and unigram counts if the trigram counts are greater than zero. If the trigram counts are zero, bigram and unigram probabilities are considered. However, in deleted interpolation strategy, lower order n-grams are also taken into account and all three models are interpolated [8]. Deleted interpolation strategy helps reducing the number of low probability n-grams.

Base recognition unit is another aspect of an ASR system which is also addressed by language modeling. The most widely used approach is taking words as the base recognition units. However, word-based models cause high OOV word rates for agglutinative languages like Turkish, Czech, Hungarian, Finnish and Korean [6]. Since agglutinative languages have productive morphologies, smaller recognition units are used in order to address the problem of high OOV word rates [5, 10, 11, 12]. Syllable based, morph based and stem-ending based word models are used in order to handle the high OOV word rate problem in agglutinative languages.

In a syllable based word model, the base recognition units are syllables. In a morph based word model, morphs are taken as the base recognition units. A morph is a statistically derived unit. In the stem-ending model, all words are morphologically parsed into their stems and endings, which are treated as the base recognition units.

2.4.1 Language Model Interpolation

Apart from using a standalone model, several language models can be unified into a combined language model in order to achieve better performing models. By using different text corpora, one can build different n-gram language models and apply language model interpolation in order to obtain a mixed language model. By using linear language model interpolation, the weighted n-gram probabilities of the given language models are computed. The resulting combined language model consists of weighted n-gram probabilities obtained from the input language models (Hsu, 2007). When interpolating two language models, an interpolation constant can be given to indicate a different weight value for one of the language models.

In (Arısoy and Saraçlar, 2006), interpolating a Turkish word-based language model with a stem-based language model was shown to improve the recognition performance of an LVCSR system. In our research, we also employed this technique in constructing word-based n-gram language models.

2.5 Decoder

The aim of the decoder in an ASR system is to find out the most likely word sequence which corresponds to a given acoustic observation. The decoder computes the argmax part in Equation 2.3. The decoder searches through all the words in the vocabulary in order to find the most likely word which corresponds to the acoustic input. In order to reduce the search space, the decoder utilizes the acoustic and the language models. With the aid of these models, the decoder finds out the best matching word string.

The recognition performance of the decoder depends on several aspects. Words that are not contained by the recognition vocabulary are deemed as OOV words and the decoder has no possibility of recognizing these words. Also, weak acoustic and language models degrade the performance of the decoder dramatically. The largeness of the search space is also an important factor which affects the performance of the decoder.

2.6 Recognition Performance Measures

The performance of an ASR system is evaluated by considering the hypothesis words (words which are produced by the recognizer) and the reference words.

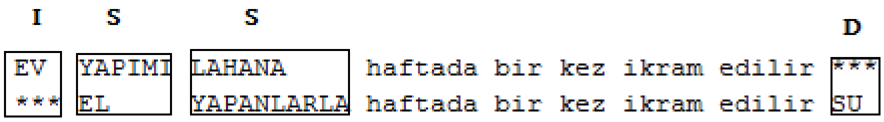


Figure 2.3 - Obtaining the reference sentence by using the hypothesis sentence

In order to reach the reference sentence by using the hypothesis sentence, a number of insertions, deletions and substitutions must be performed on the hypothesis sentence. Consider the reference sentence, “Ev yapımı lahana haftada bir kez ikram edilir” and the hypothesis sentence, “El yapanlarla haftada bir kez ikram edilir su”. In order to obtain the reference sentence; one insertion, one deletion and two substitutions must be performed (see Figure 2.3).

There are four main measures used to evaluate the recognition performance of an ASR system:

- Word Error Rate (WER)
- Correct Word Rate (CWR)
- Word Accuracy Rate (WAR)
- Sentence Error Rate (SER)

WER is calculated by the formula:

$$\mathrm{WER} = \frac{\mathrm{D} + \mathrm{S} + \mathrm{I}}{\mathrm{N}} \times 100 \tag{2.4}$$

In Equation 2.4, "D" denotes the total number of deletions, "S" denotes the total number of substitutions, "I" denotes the total number of insertions and "N" denotes the total number of words to be recognized.

CWR is calculated by the formula:

$$\mathrm{CWR} = \frac{\mathrm{N} - \mathrm{D} - \mathrm{S}}{\mathrm{N}} \times 100 \tag{2.5}$$

WAR is calculated by the formula:

$$\mathrm{WAR} = \frac{\mathrm{N} - \mathrm{D} - \mathrm{S} - \mathrm{I}}{\mathrm{N}} \times 100 \tag{2.6}$$

SER is calculated by the formula:

$$\mathrm{SER} = \frac{\text{number of correct sentences}}{\text{total number of sentences}} \times 100 \tag{2.7}$$

In this thesis, we utilized the WER, CWR and WAR metrics in our experiments since these are the most commonly used metrics in the speech recognition domain.

CHAPTER 3

SPEECH RECOGNITION IN TURKISH LANGUAGE

In this chapter, we will explain the essential morphological characteristics of the Turkish language. Then we will outline the challenges of Turkish speech recognition. Finally, the related work on Turkish speech recognition will be discussed.

3.1 Basic Turkish Morphology

Turkish is an agglutinative language. In agglutinative languages, many words can be generated from a single stem by using several suffixes [13]. The suffixes in Turkish are categorized as inflectional and derivational. Derivational suffixes differ from the inflectional suffixes in the sense that when affixed to a word, derivational suffixes may change the syntactic category of the word. However, inflectional suffixation does not change the grammatical category of a word. Some examples for derivational and inflectional suffixation are given below:

Derivational suffixation:

- Bak+ım+ lı *(well-cared for)*

  (Verb to Noun to Adjective)

- Dön+gü *(loop)*

  (Verb to Noun)

Inflectional suffixation:

- Bahçe+ler *(gardens)*

  Garden + PL

- Koş+uyor+lar+dı *(they were running)*

  Run + Progress + P3PL + Past

We can observe the effect of derivational suffixes by studying our first example, “bakımlı”. “Bakmak” *(to care)* is a verb. By affixing the derivational suffix “-ım”, we obtain the noun “bakım” *(care)*. Finally, the derivational suffix “-lı” transforms the noun into the adjective, “bakımlı” *(well-cared for)*.

Unlike derivational suffixes, inflectional suffixes do not alter the grammatical category of a word. Inflectional suffixes are used to indicate grammatical concepts such as number, person, gender, tense, aspect and modality [6].

In order to have an idea for how complex derivational suffixation can become in Turkish, it may be significant to study the example below [5], which can be translated as *“as if you were one of those whom we might consider not converting into an Ottoman”*:

osmanlılaştıramayabileceklerimizdenmişsinizcesine

The word can be decomposed as:

osman+lı+laş+tır+ama+yabil+ecek+ler+imiz+den+miş+siniz+cesine

One of the main characteristics of the Turkish language is that the language obeys the vowel harmony rule. The vowel harmony rule requires the first vowel of the suffix to comply with the last vowel of the stem. This results in the modification of some of the letters in the original word. Consider the Turkish word “Başlıyor” *(it is* *starting).* The word can be decomposed as “Başla+ıyor”. In order to satisfy the vowel harmony rule, the vowel “a” in the stem “Başla” is removed.

Free word order is another important characteristic of the Turkish language. The positions of the words in a sentence can be changed without losing the original meaning of the sentence. Although the most commonly used word order type is Subject-Object-Verb in Turkish language [2], other types of word orders are also widely used.

The main purpose of free word ordering in Turkish is to emphasize a word. The examples below demonstrate the usage of free word ordering in order to emphasize different words:

Geçen yıl Almanya’ya gittim	*(Last year I went to Germany)*
Almanya’ya geçen yıl gittim	*(It was last year that I went to Germany)*

3.2 Challenges of Speech Recognition in Turkish

The productive morphology of the Turkish language is an important problem to address in Turkish ASR systems. Since Turkish language is highly agglutinative, language modeling must be handled differently than non-agglutinative languages like English [5]. Many unique words can be produced from a single world by using several suffixes, which results in the vocabulary growth problem [2].

OOV (Out-of-Vocabulary) words are important causes of high error rates in ASR systems. For an ASR system, it is impossible to recognize a word which is marked as OOV. Since Turkish has a highly productive morphology, it is very hard to cover all the words which can be derived from a single stem. This property of the language makes it easy to have OOV words in Turkish ASR systems. Vocabulary sizes which are considered large for English language do not suffice for Turkish in terms of OOV rates [2].

In order to address the problem of high OOV rates in morphologically rich languages, it is proposed to generate language models by using sub-word units instead of words. By using sub-word units, it is aimed to reduce the vocabulary size. Stem/Endings and morphs are the most commonly used sub-word units in ASR systems which aim to handle the OOV problem in agglutinative languages. Sub-units can be obtained by using morphological parsing and statistical principles. Language models generated by using sub-words require higher order n-grams than words [2].

Although language models generated by using sub-word units address the OOV rate problem, they introduce the possibility of generating non-word outputs. Such language models do not guarantee rules like vowel harmony [2].

N-grams are used for estimating the next word in a sentence with a probability. The larger the number of candidate next words, the less robust the language model is. Since Turkish language allows free word order, it is also difficult to obtain robust language models for the language.

3.3 Related Work on Turkish LVCSR

Several researches have been made about Turkish ASR systems. Most of the studies investigate the results of employing different recognition units. [5] is one of the first studies which proposes to generate the language model for Turkish by using units smaller than a word. In [6], alternatives to the word-based model are investigated. In this research stems, endings and morphemes are used. Also, a language model which combines words, stems, endings and morphemes was proposed. Stem/ending based model is referred as a solution to address the trade-off between the small coverage caused by word-based model and lack of acoustic information introduced by the morpheme based model [6]. However, best recognition results are obtained by using the word-based model. In [7], word-based language model is compared against morpheme-based, stem/ending based and syllable based language models. Best recognition results are obtained by using the word-based model. Syllable based model is reported to give the worst recognition results. The size of the text corpus utilized in this research is reported to be insufficient for bigram modeling. In [4], stem/ending based language model is compared with word-based language model. Best recognition results are obtained by using the word-based model. In these researches, bigrams were employed in language models.

In [2], stem/ending and morph-based language models are compared to word based language model. 3-grams and 4-grams are utilized in this research. Stem/ending based model is shown to outperform the word-based model by 0.6%. Also, morph-based model is reported to outperform the word-based model by 0.8%. In that study, best recognition results are obtained by using the morph-based model. However, it is important to note that the amount of audio training data utilized in this research is quite large (approximately 194 hours, [2]) with respect to other researches held in Turkish LVCSR area.

3.4 Related Work on Data Selection

The ability of a learning system is dependent on the quality of the learning data as well as the learning methods [14]. Apriori information can be utilized to select training data in learning based systems. When large amounts of training data is available, a subset which represents the entire set best is required if the computational cost of processing the entire training data is not feasible. In [15], an MLP-based feature extraction approach was proposed for ASR systems. In that research, it is reported that, by selecting the 60% of the entire training data, almost the same recognition results were obtained with respect to the system which was trained by using the entire training data. Data selection is also utilized in ASR systems if acquiring a transcribed, rich audio corpus is expensive. In [16] and [17], methods to improve the error rate without incurring additional transcription cost were proposed. In [16], two selection approaches based on error rate value were introduced: selection by low recognition error and selection by high recognition error. The latter approach, in which the utterances with higher recognition errors were doubled in the training data, yielded to a higher gain in recognition performance. In that research, it is shown that, by selecting the relatively more useful data, error rate can be improved without acquiring additional training data. In [18], by using a data selection approach, it was shown that the length of the training utterances had a significant impact on the recognition performance. It was reported that shorter utterances had approximately 50% lower accuracy than the longer utterances. The ASR system in that research was built to recognize digits in Dutch.

Data selection can be used in order to obtain more robust in-domain data distributions. In [19], a sentence is selected if introducing it into the previously selected set of sentences reduces the entropy value. By employing such an approach, models which perform better in terms of WER and perplexity were built. Another in-domain adaptation improvement was obtained in [20] by artifically generating training data by considering vocal tract length. This approach is motivated by the fact that many triphones in small training sets are likely to be spoken by few speakers.

CHAPTER 4

DATA AND TOOLS FOR TURKISH LVCSR

In this chapter, we explain the content of the audio and text data used to train and test our system. Also, details about the tools used for acoustic model and language model generation and decoding are given.

Since speech recognition requires combining the outputs of several applications, it can be difficult for a novice to decide how to start studying speech recognition. This chapter also aims to provide a brief guidance about speech recognition tools for the relatively inexperienced readers.

4.1 Acoustic Data

We used the METU Turkish Microphone Speech corpus [21] in our experiments.

120 speakers (60 male and 60 female) speak 40 sentences each (approximately 300 words per speaker). The 40 sentences are selected randomly for each speaker from a triphone-balanced set of 2462 Turkish sentences. The speakers are selected from students, faculty and staff at METU and all are native speakers of Turkish. The age range is from 19 to 50 years with an average of 23.9 years [21].

The audio corpus contains 4769 sentences along with transcription files for the sentences.

It is significant to note that acquiring a Turkish audio corpus with sufficient size is quite difficult. We do not know any publicly available, phonetically-balanced audio corpus for Turkish language other than the METU Turkish Microphone Speech corpus.

The size of this audio corpus is very limited with respect to audio corpora for other languages. The content of this audio corpus is also very limited with respect to the size of the acoustic data utilized in [2].

4.2 Text Data

In this research, we utilized three different text corpora of different sizes in order to generate n-gram language models.

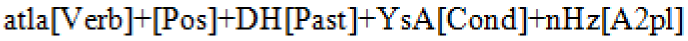
We categorized the text corpora with respect to their sizes as *small*, *medium* and *large*. The small corpus is composed of the transcriptions of the utterances in the audio corpus. The content of the small corpus can not be categorized into a specific domain. The medium and the large corpus consist of sentences from the news domain. The large corpus utilized in this research is publicly available at [22]. In Table 4-1, some numerical details about the text corpora are listed.

Table 4-1 - Text Corpora Details

Corpus	# of sentences	# of words	# of unique words
Small	4769	33800	7361
Medium	442947	4659902	218382
Large	9831256	91789908	1169127

4.3 Morphological Parser

Morphological parsers are used to decompose word into their stems and morphemes. In this research, we utilized the morphological parser developed by Haşim Sak. This morphological parser is available at [22]. The output of this parser consists of possible decompositions of the input words. We utilized the parser in generating stem/ending based language models. The endings are obtained by merging the morphemes provided by the parser. An example decomposition of this parser is given in Figure 4.1 for the Turkish word “atlasaydınız”.



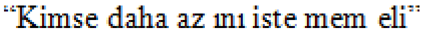
atla[Verb]+[Pos]+DH[Past]+YsA[Cond]+nHz[A2pl]

Figure 4.1 - Decomposition of the Turkish word “atlasaydınız”

4.4 Morfessor

Morphs are morpheme-like units which are obtained by taking into account the statistical properties of words in a text corpus. Morphs are statistically derived units and they are generated regardless of the morphological structure of the language.

We used the Morfessor tool [23] in order to obtain morph decompositions, which is freely available. The Morfessor tool applies an unsupervised learning approach in decomposition. An example sentence is which is decomposed into its morphs is given at Figure 4.2.



“Kimse daha az mı iste mem eli”

Figure 4.2 - Morph decomposition of a Turkish sentence

4.5 Hidden Markov Model Toolkit (HTK)

We utilized HTK [24] to generate acoustic models. HTK is a widely used tool in speech recognition applications. The library provides tools to generate HMM-based acoustic and language models, as well as decoding and evaluation tools. There is a freely available, complete guide named “HTK Book” which consists of comprehensive details about how to use the tools in the library.

4.6 Stanford Research Institute Language Modeling Toolkit (SRILM)

The SRILM toolkit [25] is another widely used toolkit for speech recognition purposes. We used SRILM in order to generate n-grams for language modeling and perplexity calculations.

4.7 Julius

Julius is an open-source LVCSR engine [26]. We utilized Julius as the decoder in our experiments. The recognitions results are also obtained by using Julius.

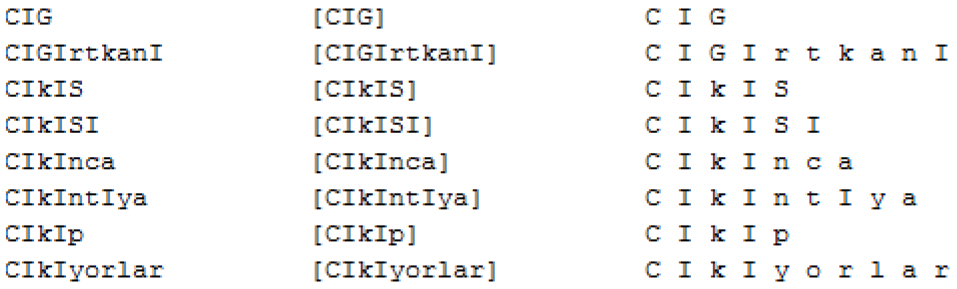
CHAPTER 5

APPROACHES USED IN THIS RESEARCH

In this chapter, we will outline the details of the approaches we employed in order to solve the problem of Turkish LVCSR. First, we will give the details of the acoustic model generation. Next, we will explain the methods we used in generating the language models. Finally, we will outline the details of the decoding optimization process.

5.1 Acoustic Model Generation

In acoustic model generation, the initial step is to create a pronunciation dictionary. The pronunciation dictionary consists of the words in the speech corpus and their pronunciations. An excerpt from the dictionary used in this thesis is given in Figure 5.1.



```
CIG                 [CIG]               C I G
CIGIrtkanI          [CIGIrtkanI]        C I G I r t k a n I
CIkIS               [CIkIS]             C I k I S
CIkISI              [CIkISI]            C I k I S I
CIkInca             [CIkInca]           C I k I n c a
CIkIntIya           [CIkIntIya]         C I k I n t I y a
CIkIp               [CIkIp]             C I k I p
CIkIyorlar          [CIkIyorlar]        C I k I y o r l a r
```

Figure 5.1 - An excerpt from the pronunciation dictionary

We utilized the HTK tool for acoustic model generation. Since this tool does not allow Turkish characters, we replaced each Turkish character with an uppercase latter of the English alphabet. The first line the pronunciation dictionary contains the pronunciation of the word “çığ”. Since the spellings of Turkish words are very similar to their spoken representations [2], we generated the pronunciation dictionary by using the 29 letters of the Turkish alphabet.

The next step is to convert the speech signals into feature vectors. In this research, we used Mel Frequency Cepstral Coefficients (MFCC), which are commonly used in obtaining feature vectors from the raw speech signal. We used the HCopy program to convert the .wav files into their corresponding feature vector representations.

After obtaining the dictionary and the feature vector representations of the speech files in the corpus, we created the monophone HMMs. The purpose of this stage is to model the phones used to represent the words in the pronunciation dictionary. This stage was completed by re-estimating the monophones by using the HERest program of the HTK library.

In LVCSR systems, it is required to handle the short pauses that exist between utterances. These pauses are longer with respect to end-of-sentences pauses. In order to handle short pauses, we introduced an extra model into our monophone models by using the HHed and HERest programs of the HTK library. By the end of this stage, the short pause model and the long silence model are tied.

After generating the silence model, we realigned the training data. The purpose of this operation is to consider all pronunciations of a word in the training data and choose the pronunciation which represents the acoustic data best. This operation was completed by using the HVite and HERest programs.

Considering the neighbour phones is an important task in acoustic model generation. The motivation behind this task is to model the phones in their contexts by considering the immediate left and right phones in order to obtain a more accurate representation for the phones. We obtained the triphone models by using the HDMan, HHed and HERest programs of the HTK library.

The final step in acoustic model generation is referred as creating tied-state triphones. The motivation behind this step is to tie the similar acoustic states in the context dependent triphones in order to obtain more robust estimations.

5.2 Language Model Generation

In this research, we used the word based, stem-ending based and morph based approaches in generating the language models. In this section, we will outline the methodologies used to generate and enhance these approaches. The results obtained by these methodologies are given in Chapter 6.

5.2.1 Word Based Language Modeling

Word based language models are generated by considering the words as recognition units. Unlike morphological and statistical parsing strategies, word based strategy requires to consider the words as they are (i.e., unparsed). The n-gram models are generated by taking words as the recognition units.

We extended the traditional word based model by integrating two approaches into our word based models. The first approach is based on merging words that are frequently seen together in the text corpus. The second approach is based on creating an extra language model by using the stems of the words and interpolating the stem based language model with the word based model.

Although written separately in several texts, some words in Turkish are frequently used together in many contexts. We merged such words in order to obtain more robust n-gram estimates. This approach was also employed in [6]. We merged the words that satisfy one of the criteria below:

1. The words must be written together in accordance with the Association of Turkish Language [27].

2. The words must be used together frequently in many contexts.

In [6], the maximum size for the compound word was taken as 10. In [28], the maximum size for a Turkish verb is given as 7.6 on average. In this research, we imposed the restriction that the length of the compound word should not exceed 10 characters. The word pairs which were merged with respect to these conditions are given in Appendix A.

The second enhancement of the word based model is related to stem based modeling. Stem based modeling is shown to lead to better language model estimations in [4, 29]. In this approach, we used a morphological analyser in order to obtain the stem parts of the words. Next, we generated the n-grams probabilities by using the stem parts. Finally, the stem based and the word based language models were interpolated into a mixed language model. The algorithm for our enhanced word based language modeling is given below.

Enhanced Word Based Language Modeling:

1. Gather all word pairs <w1, w2> which satisfy the following criteria:
    - Words must be written together in accordance with Turkish rules
    - Words that are observed together frequently
2. For all word pairs <w1, w2> obtained in step 1, if the length of the resulting word obtained by merging w1 and w2 is equal to or less than 10, merge w1 and w2.
3. Generate a 3-gram, word-based language model (WLM)
4. Obtain the stem parts of every word in the text and audio corpus.
5. Generate a 3-gram, stem-based language model (SLM)
6. Interpolate models WLM and SLM into a unified language model.

5.2.2 Stem-Ending Based Language Modeling

In stem-ending based language models, stems and endings are treated as recognition units. Hence, the n-gram probabilities are calculated by using stems and endings. For the stem-ending based model, we obtained the stems parts of the words by using the morphological analyser and kept the remaining parts of the words as endings.

We modified the stem-ending model based on three approaches. The first approach is the one outlined in the previous section, namely, merging the word pairs that are observed together frequently in training data. The second approach is based on merging stems and endings which consist of a single letter. This approach was also utilized in [4]. The third approach is based on choosing the morphological parsing which yields to longest stem parts. The algorithm for the extended stem-ending based language modeling used in our research is given below.

Enhanced Stem-Ending Based Language Modeling:

1. Parse all of the words in the text and speech corpora using the morphological parser.

2. Among all parses of each word, choose the decomposition that yields to longest stem.

3. Merge the words that are observed frequently.

4. Merge single-letter stems and endings.

5. Generate a 4-gram language model by using the processed stems and endings.

5.2.3 Morph Based Language Modeling

Morphs are statistically derived sub-word units, produced by an unsupervised language modeling algorithm. We used the Morfessor tool in order to obtain the morph decompositions of the words. The only additional approach we employed in morph based language modeling is the merging of word pairs observed together frequently.

5.2.4 Hybrid Language Modeling

We introduced a hybrid language model which incorporates word based and stem ending based language models. In this approach, we left the most frequent 10K words (out of ~1.2 M unique words) as unparsed. The rest of the words in the text and speech corpora were parsed into stems and endings. The algorithm for our hybrid language modeling is given below.

Hybrid Language Modeling:

1. Find out the most frequent 10K words in the text/speech corpora and leave them as unparsed.

2. Parse the rest of the words into their stems and endings by in accordance with the algorithm given in section 5.2.2.

3. Generate a 4-gram language model based on words, stems and endings.

5.3 Data Selection Approaches

As pointed out earlier, it is difficult to obtain a rich audio corpus for the Turkish language. In this research, we utilized a limited audio corpus to generate the acoustic models. In order to obtain more robust acoustic models, we developed several approaches mainly based on the analysis and reorganization of the audio training data. We introduced three approaches in order to improve the acoustic model generated by using the limited audio corpus. The first method is based on an empirical approach which attempts to find out a normalized distribution in the audio corpus by adjusting the number of utterances in the corpus. In the second approach, we duplicate the utterances with lower perplexity values. The third approach is based on duplicating the utterances with high WER values.

5.3.1 Empirical Data Selection

By analyzing the utterances in the audio corpus, we found out that some sentences are uttered by more than one speaker. In this method, we attempted to find a normalized configuration for the audio corpus by adjusting the number of utterances based on their occurrence counts. We conducted experiments in which we duplicated utterances with lower occurrence counts and removed samples of utterances with higher occurrence counts.

5.3.2 Perplexity Based Data Selection

In this approach, we calculated the perplexity of every utterance in the training audio corpus and calculated an average utterance perplexity value. We duplicated the utterances with lower perplexity values in the training audio corpus. By increasing the number of utterances with lower perplexity values, we aimed to increase the weight of the more "probable" sentences in the acoustic model.

5.3.3 Word Error Rate (WER) Based Data Selection

In this method, we attempted to reduce the WER values by duplicating the utterances with high WER values in the training audio corpus. For this approach, we first generated an acoustic model by using the training utterances. Then, we conducted a recognition experiment in which the acoustic model was tested by using the training utterances. By analysing the recognition results, we duplicated the training utterances with higher WER values.

5.4 Optimization of Decoding Parameters

We investigated the optimal values for the decoding parameters, "beam width" and "language model penalty". See Appendix B for the detailed definitions of these parameters taken from the Julius manual. We employed a brute force approach in order to investigate the optimal values of the decoding parameters incrementally. The algorithm for the beam-width parameter is given below. The optimal value for language model penalty is investigated by using the same approach.

For beam-width values in [700, 5000], execute the steps below:

1. Record start time.
2. Run the decoder by using the test data.
3. Calculate the processing duration.
4. Calculate and store speech recognition measures (CWR, WER, WAR) for the current beam-width values.
5. Increment beam-width value by 10.

We similarly investigated the optimal value for the language model penalty parameter. We also conducted an additional optimization experiment in which the optimal values for the beam-width and language model penalty parameters were investigated together.

CHAPTER 6

EXPERIMENTAL RESULTS & DISCUSSION

In this research, we experimented with several ASR concepts in order to investigate which configuration gives the best recognition results by using a limited audio corpus. Our experiments are mainly based on language modeling.

Language modeling experiments include:

- Generating n-grams with different orders by using text corpora with different sizes
- Using different kinds of recognition units: word, stem/ending and morph
- Using an alternate pronunciation dictionary which considers phoneme relations

Also, we investigated the optimal values for the following decoding parameters:

- Beam width
- Language model penalty

Since the acoustic data utilized in our research is limited, we reorganized the test data in order to include sentences uttered by more than one speaker. However, we did not include these sentences in the language models.

We also conducted a recognition experiment in which the language model contains the sentences utilized in testing the system. The motivation behind this experiment is to find out the lowest WER value bound for the system.

We also experimented with a hybrid model which is based on word-based and stem/ending based n-grams. In this experiment, we chose to leave the most frequent 10K words unparsed.

6.1 The Recognizer

We designed the recognizer used in this research by using HTK, SRILM and Julius tools. The properties of the recognizer are given as:

- The acoustic model component of the recognizer is trained by using the HTK tool. 95% of the sentences in the audio corpus are used for training and 5% is used for testing (refer to Table 6-1 for details). Each acoustic model in our experiments is based on tied-state triphone HMMs, which is a standard approach to handle the phone relations by taking into account the neighbourhood context.
- The recordings are in .wav format, with 16 KHz sampling rate.
- Language models for different recognition units are generated by using text corpora with different sizes.
- Julius is used as the decoder component. The recognition results are also produced by using Julius.

Table 6-1 - Distribution of training/test sentences

Number of training sentences	Number of test sentences
4530	239

In the following sections, we will describe our experiments carried out with the recognizer described above.

6.2 Word-Based Recognition Results

The experiments outlined under this section are carried out by taking the smallest recognition unit as the word itself. We utilized two different text corpora with different sizes while generating the word-based n-grams. 3-grams are used for the word-based model, as suggested in [2]. Recognition results are given in Table 6-2.

Table 6-2 - Word-based recognition results

Corpus	CWR	WER	WAR
Small	46.80%	57.95%	42.05%
Large	54.33%	52.86%	47.14%

As it can be inferred from the values, generating the 3-gram model by using the large corpus yields to better recognition results.

By observing the text corpora, we found out that some words are written together or separately in different portions of the text corpora. Despite the fact that these words are used together in most of the context, they are treated as different tokens when these words are written separately. We merged the words which are used together frequently. Some examples for this case are “bir şey”, “bir kaç”, “hiç bir” and “her zaman”. After merging the words which are used together frequently, we obtained better recognition results which are outlined in Table 6-3.

Table 6-3 - Recognition results with improved stem-ending model

CWR	WER	WAR
55.77%	50.50%	49.50%

Finally, we generate a stem-based language model and interpolate it with the word based model. As pointed out in [29], combining different language models with interpolation weights can yield to more robust language model estimations. The best recognition result was obtained by assigning the weight of the word based language model as 37%. The interpolation operation was carried out by using the “-mix-lm” parameter of the “ngram” program, which is available in the SRILM toolkit. As it can be inferred from Table 6-4, we obtained a 0.32% decrease in WER value.

Table 6-4 - Results after interpolating stem-based & word-based models

CWR	**WER**	**WAR**
56.22%	50.18%	49.82%

6.3 Stem-Ending Based Recognition Results

In order to generate stem-ending based language models, we first parsed the entire text corpora and transcription files by using Sak's morphological parser [22]. Since the output of the parser contains detailed morpheme information, we selected the stem part of the decomposition and treated the rest of the word as the ending part. We replaced every word in the text corpora with its corresponding stem and ending part. Then we generated the language models by using the processed text corpora.

We experimented with three different n-gram orders for the stem-ending based model. 3-gram, 4-gram and 5-gram models were used with respect to text corpus size. The 3-gram model was generated using the small corpus. The 4-gram and 5-gram models were generated using the large corpus. The recognition results for different n-gram orders are given in Table 6-5.

As it can be inferred from the recognition results, the 4-gram language model yields the best performance. Recognition performance with the 5-gram language model is slightly worse than the performance with the 4-gram language model.

Table 6-5 - Stem-Ending based recognition results

Corpus	**N-gram**	**CWR**	**WER**	**WAR**
Small	3	41.03%	61.68%	38.32%
Large	4	42.92%	59.53%	40.47%
Large	5	42.87%	59.58%	40.42%

By observing the recognition results, we found out that stems and endings which consist of a single letter degrade the recognition performance. Since one-letter long stems and endings convey rather insufficient acoustic information, they increase the probability of words to be confused acoustically. Some of the improvements on words basis are given in Table 6-6.

Table 6-6 - Some improvements obtained by merging single-letter stems and endings

Before Merge		After Merge	
Hypothesis	Reference	Hypothesis	Reference
o nun	otla nın	onun	Onun
kaz ı	kazı	kazı	Kazı
iş i	Işi	işi	Işi

Also, instead of choosing a random decomposition for the word, we preferred to use the decomposition which contains the longest stem. It is important to note that we did not apply disambiguation to choose the best fitting decomposition, since it is reported in [31] that choosing disambiguated decompositions do not perform well for stem-ending based models.

The combined effect of merging single-letter stems and endings, choosing the longest stem and merging the words which are used together frequently is given in Table 6-7.

Table 6-7 - Recognition results with improved stem-ending model

CWR	**WER**	**WAR**
48.87%	55.04%	44.96%

As it can be inferred from the results, we obtained better recognition performance with our improved stem-ending model.

6.4 Morph-based Recognition Results

We experimented with 3-gram and 4-gram language models for the morph-based model. The 3-gram and 4-gram language models were generated by using the medium and the large text corpus respectively. Recognition results with the morph-based model are given in Table 6-8.

Table 6-8 - Morph-based recognition results

Corpus	N-gram	CWR	WER	WAR
Medium	3	40.02%	61.10%	38.90%
Large	4	53.65%	51.71%	48.29%

As it can be inferred from the results, the recognition performance obtained by using the large text corpus is significantly higher. We observed the morph decompositions for medium-corpus based and large-corpus based models. In the first and second experiments, each word is segmented into 1.2 and 1.1 units on average. The 4-gram language model generated by using the large corpus contains longer morphs. Longer morphs and higher order n-gram provide the significant increase in the recognition performance.

6.5 Hybrid language model

In our experiments, we obtained better recognition results with longer recognition units. In this experiment, we attempted to combine the word-based and the stem-ending based approach.

The large text corpus utilized in this research consists of ~1.2M unique words. We left the most frequent 10K unparsed. The rest of the words are parsed into their stems and endings. We generated a 4-gram language model by using the large text corpus. The recognition results are given in Table 6-9.

Table 6-9 - Recognition results obtained by using the hybrid language model

CWR	WER	WAR
49.38%	54.72%	45.28%

The recognition results of our hybrid language model are worse than the results obtained by using the word-based model and are slightly better than the results obtained by using the stem-ending based model.

This experiment also supports the argument that using longer recognition units yields to better recognition performance with our present acoustic and language model. We claim the main reason of this situation to be our non-robust acoustic model which is built by using a limited audio corpus.

6.6 Comparison of Results with Different Recognition Units

Among our experiments with different recognition units, we observe the best results by using word-based language models. The results obtained by using different recognition units are given in Table 6-10.

Table 6-10 - Recognition results with different recognition units

Recognition Unit	**CWR**	**WER**	**WAR**
Word	55.77%	50.50%	49.50%
Stem-Ending	48.87%	55.04%	44.96%
Morph	53.65%	51.71%	48.29%

Due to the highly productive nature of the Turkish language, it is very likely to have OOV words by using the word-based model. By using sub-word units, it is attempted to achieve higher word coverage over the test data. However, there is a trade-off between obtaining higher word coverages by using sub-word units and the lack of acoustic information introduced by using smaller recognition units.

We calculated the number of OOV words in the test data for different recognition units. As depicted in Table 6-11, sub-word units reduce the number of OOV words and increase the total number of words in the test data.

Table 6-11 - Test data statistics with respect to recognition unit

Unit	# of sentences	# of words	# of unique words	OOV Rate
Word	239	1724	1274	3.4%
Stem-Ending	239	2358	1349	2.1%
Morph	239	1798	1326	1.2%

As it can be inferred from Table 6.11 and our recognition results, there exists a correlation between the number of unique words and WER values.

We also calculated the perplexity values of the language models over the test data. We took into account the cost of the OOV words in perplexity calculation. Intuitively, perplexity can be considered as the weighted average number of choices a random variable has to make [8]. If the perplexity of a language model is high, the recognizer has to consider a lot of words in order to determine the next word. Therefore, low perplexity values are desired. However, low perplexity values do not yield to high recognition results every time [6]. Total log probabilities and perplexity values are given in Table 6-12.

Table 6-12 - Total log probabilities and perplexity values

Recognition Unit	Total log probability	Perplexity
Word	-7588.7	871.4
Stem-Ending	-7640.7	379.8
Morph	-7362.8	2707.4

It can be inferred from Table 6-12 that the stem-ending based model has the lowest perplexity value. However, as stated previously, low perplexity values do not yield to high recognition results every time, as in our case.

We consider the main reason of the relatively poor recognition results obtained by using sub-word units to be the insufficient amount of acoustic data used in our research. Sub-word based models aim to address the small coverage problem. However, sub-word based approaches also introduce smaller recognition units, which may not contain sufficient acoustic information. Therefore, in order to observe the actual contribution of sub-word based models, we need more robust acoustic models which are trained by using sufficient amount of speech data.

6.7 Using an Alternate Pronunciation Dictionary

In Turkish, some vowels and consonants are pronounced differently depending on the place they are produced in the vocal tract. We utilized the program used in [21] to replace Turkish letters with the corresponding characters in the SAMPA alphabet.

We obtained 44 distinct phonemes. In our previous experiments, we used the 29 letters of the Turkish alphabet as phonemes.

We generated a 3-gram word based language model by using the small text corpus. We obtained a WER value of 62.84%, which is worse than the recognition result (WER=57.95%) obtained by using the standard (i.e., solely consisting of the 29 letters of the Turkish alphabet) pronunciation dictionary. We may not have observed the advantage of this dictionary since SONIC tool was used in [21]. In [21], the speaker adaptation and normalization capabilities of SONIC were reported to be used.

6.8 Reorganization of Test Data Due to Lack of Audio Corpus

In this research, we made use of an audio corpus of ~5.2 hours only. Compared to [2], the amount of audio data used in our research is very insufficient.

Because of such insufficiency, we decided to reorganize our training and testing data in such a way that the test audio corpus consists of sentences uttered by more than one speaker. For example, if a sentence is uttered by 5 speakers in the entire audio corpus, 4 of the utterances are kept for acoustic training and the remaining one utterance is introduced into the test corpus. It is important to note that we did not include these sentences when generating the language model.

In order to accomplish this task, the sentences which are uttered by more than one speaker are investigated. The distributions of sentences in the audio corpus are given in Table 6-13. We built the test corpus as to include the 5% portion of the audio corpus by choosing from the sentences which occur the most (i.e., starting to choose from sentences which occur 9 times, then from sentences which occur 8 times and so on). We generated a 3-gram, word-based language model for this experiment, which gives the best results among our experiments. The 3-gram model was generated by using the large text corpus. Refer to Table 6-14 for recognition results.

Table 6-13 - The distributions of sentences in the audio corpus

Occurence count of a sentence (N)	How many sentences exist that have an occurence count of N?
1	848
2	653
3	387
4	205
5	77
6	35
7	3
9	2

As it can be inferred from Table 6-14, by reorganizing test data, word-error rate has dropped to 47.72% from 52.86%.

Table 6-14 - Recognition results with reorganized test data

CWR	WER	WAR
57.92%	47.72%	52.28%

By observing the recognition results, we found out cases where a frequent sentence is recognized with poor accuracy. Some of the reasons to such poor results are found to be:

- Not pronouncing every sound of the word properly
- Accent differences between speakers
- Low probabilities for words in the language model

After this experiment, we also conducted a cheating experiment in which the test sentences already exist in the language model. The motivation behind this experiment is to find out the possible lowest WER value bound for the system. Recognition results obtained by using the cheating test data are given in Table 6.15.

Table 6-15 - Recognition results obtained by using the cheating test data

CWR	WER	WAR
78.68%	25.11%	74.89%

6.9 Data Selection Experiments

We employed three different approaches in order to improve the robustness of the acoustic model by considering the distributions, perplexities and WER values of the training utterances.

In the empirical approach, we adjusted the number of utterances in the training corpus in accordance with their occurrence counts. The recognition results obtained by the empirical approach are listed in Table 6-16. We obtained a 0.27 decrease in WER when the occurrence count of all utterances were adjusted to three. This improvement is statistically significant at the level of $p < 0.001$ as measured by the NIST SIGN test.

Table 6-16- Empirical data selection recognition results

Method	**WER**
Baseline	50.50%
All sentences occur once	55.68%
All sentences occur twice	52.23%
All sentences occur three times	50.23%
All sentences occur four times	51.00%
All sentences occur five times	50.77%
All sentences occur six times	51.63%
Sentences occurring 1-3 times raised to 4	51.18%
Sentences occurring 5-9 times reduced to 4	50.36%
Sentences occurring 5-9 times reduced to 3	50.41%
Sentences occurring 4-9 times reduced to 2	51.86%

In the perplexity based approach, utterances with lower perplexities are duplicated in the training data. We experimented with different average perplexity boundaries. The recognition results obtained by the perplexity based approach are listed in Table 6-17. No improvement over the baseline system was observed.

Table 6-17 - Perplexity based data selection recognition results

Method	WER
Baseline	50.50%
Duplicate sentences with perplexities lower than average perplexity	51.00%
Duplicate sentences with perplexities in the best 50% segment	50.86%
Duplicate sentences with perplexities in the best 25% segment	50.77%
Reduce sentences with perplexities higher than average perplexity	50.95%
Reduce sentences with perplexities in the worst 50% segment	53.27%

The recognition results obtained by the WER based approach are listed in Table 6-18. We obtained a 1.77% decrease in WER when the training utterances with WER values higher than 50% were duplicated 4 more times in the audio corpus. This improvement is statistically significant at the level of $p < 0.001$ as measured by the NIST SIGN test.

Table 6-18 - WER based data selection recognition results

Method	WER
Baseline	50.50%
Copy sentence 2 more times if WER > 50%	50.23%
Copy sentence 3 more times if WER > 50%	48.96%
Copy sentence 4 more times if WER > 50%	48.73%
Copy sentence 4 more times if WER > 37.5%	49.68%
Copy sentence 4 more times if WER > 62.5%	50.27%
Copy sentence 5 more times if WER > 50%	50.86%
Copy sentence 6 more times if WER > 50%	51.41%

6.10 Cross-validation of WER-based Data Selection Approach

We tested our best performing data selection approach (Copy sentence 4 more times if WER > 50%) by using three additional training and testing data sets. We obtained the data used in these experiments by shuffling the data in our main training and testing sets. There is no overlap between the training and testing data with respect to the language and acoustic models utilized in these experiments. We obtained a decrease of 0.41%, 1.27% and 1.43% in WER values.

By considering our original test results along with these additional experiments, we obtained an average WER=52.2% for the baseline and an average WER=50.9% for our improved model, yielding to an average 1.3% decrease in WER value.

These cross-validation experiments statistically indicate that our best performing WER-based data selection approach improves the recognition performance.

6.11 Optimization of Decoding Parameters

We chose the 3-gram, word-based recognizer (WER=47.72%) as our baseline system. In order to improve the recognition performance of our baseline system, optimal values for several decoding parameters were investigated. In this thesis, we used Julius as the decoder. We chose to investigate the optimal values for the beam-width and language model penalty parameters since investigating optimal values for these parameters are common in decoder optimization.

We employed a brute-force approach in order to incrementally investigate the optimal values for beam-width and language model penalty parameters. We selected the 5% of the sentences in the audio corpus to be used in the investigation of optimal decoding parameters.

For the beam-width parameter, we conducted a search between the values 700 and 5000, inclusively. We decided the lower bound to be 700 since the default value which Julius deems for the beam-width parameter is 800. Increasing the beam-width value yields to better recognition performance; however it also increases the processing time. At each iteration, the beam-width value is increased by 10. Since there are many values between 700 and 5000, only the values providing a significant improvement are listed in Table 6.20.

Although there is a 1% performance improvement in terms of CWR measure between beam-width=1730 and beam-width=3570 values, the average processing time also increases to 449 from 258, which is the greatest amount of increase among all values. We use the Real-time factor (RTF) metric in order to tell if this amount of time is acceptable.

Real-time factor is a metric used to measure the speed of an ASR system, given as:

$$RTF = P/I$$

Here, P is the time it takes to process an input of duration I. If RTF is equal to or less than 1, the processing is deemed to be done in real time.

By sampling different sentences in the audio corpus, the average sentence duration is calculated as 4.5 seconds. Since the test corpus for this experiment consists of 239 sentences, R is approximately equal to 1049 sentences. For beam-width=3570, the total decoding duration is 738 seconds. Hence, we have RTF=0.703. So, we can infer that our baseline recognizer satisfies the real time condition.

It is significant to note that between beam width values 4500 and 5000, no improvement on the WER value was observed. However, processing time kept increasing. So, we did not investigate the performance for beam-width values greater than 5000.

Next, we experimented with the language model parameter. For the weight factor, we experimented with the values 7.0, 8.0 and 9.0. For the penalty factor, we chose the search range as [-3.0, 0.0]. The default values that Julius assigns to weight and penalty are 8.0 and -2.0, respectively. Penalty factor is increased by 0.1 in each iteration. Best results obtained with language model penalty parameter are given in Table 6.19.

After investigating the optimal values for the beam-width and language model penalty parameters individually, we conducted a combined optimization experiment in which the optimal values for these parameters were investigated together. This experiment was conducted by using our enhanced word model and applying our best performing data selection strategy (copying a sentence 4 more times if WER > 50%). Refer to Table 6.21 for the results of this experiment.

Table 6-19 - Best results obtained with language model penalty parameter

Weight	Penalty	WER
7.0	-2.9	47.72%
8.0	-1.5	47.67%
9.0	-2.5	47.76%

6.12 Summary of Results

In this research, we experimented with several approaches existing in the literature of Turkish LVCSR by using a limited audio corpus.

We obtained the best recognition performance by using a 3-gram, enhanced word based recognizer (see 5.2.1 for the details of our enhanced word model) with optimal decoding parameters, which was incorporated with our best WER-based data selection strategy. Our performance gains are given in Table 6-22.

Table 6-20 - Recognition results for different beam-width values

Beam-Width	WER	CWR	Total Processing time (sec)	Average Processing Time Per Word (msec)
700	48.94%	57.08%	227	137
770	48.07%	57.49%	236	143
800	47.72%	57.92%	244	148
980	46.96%	58.85%	271	165
1190	45.72%	60.45%	311	189
1450	44.80%	61.69%	363	221
1730	43.82%	63.07%	425	258
3570	43.16%	64.03%	738	449

Table 6-21 - Recognition results after combined optimization

CWR	WER	WAR
61.04%	44.05%	55.95%

Table 6-22 - Performance gains obtained in our research

Approach	WER	Gain
Word based LM (1)	52.86%	Baseline
Enhanced Word LM (2)	50.18%	+2.7%
Stem-Ending based LM (3)	55.04%	-2.2%
Morph based LM (4)	51.71%	+1.1%
Hybrid LM (5)	54.72%	-1.8%
(2) with Beam-width & LM Opt.	47.14%	+5.7%
WER based data selection (6)	48.73%	+4.1%
(6) with combined optimization	44.05%	+8.8%

6.13 Comparison with Related Work

A comprehensive Phd research which addresses the challenges for LVCSR in Turkish has been held by Arısoy [2]. In this research, it has been shown that using sub-words instead of words as recognition units yields to better recognition results.

In [2], the best recognition performance is obtained by using a morph based language model, with WER=22.9%. However, in our experiments, taking words as recognition units led to better recognition results. Also WER values obtained in our research are significantly higher. However, it is not meaningful to compare our work with [2] since the amount of acoustic data utilized in Arısoy's research (approximately 194 hours, [2]) is significantly larger than the acoustic data we were able to use (approximately 5.2 hours) in this thesis.

In [6], an LVCSR system was developed for the task of broadcast dictation in Turkish. In that research, word-based model is compared with combined models, which consist of word based, stem-ending based and morpheme based approaches. 8923 utterances are reported to be used in acoustic model training. The best recognition performance is obtained by using the word-based model, yielding to a correct-word rate of 46.29% [6]. The amount of acoustic training data is more sufficient than the amount of data utilized in our research.

In [4], stem-ending based and word-based models were used. 7650 utterances are reported to be used in acoustic model training. 220 sentences, which are selected arbitrarily from the text corpus that is used to extract bigram probabilities, are used as test data [4]. The language model is obtained by using bigrams. The best recognition is obtained by using the recognition unit as word, with a CWR value of 67.86%. Refer to the Table 6-23 for the comparison of our best recognition results with the best recognition results obtained in [4]. In order to make a fair comparison, we demonstrate the results obtained by including the test sentences in the language model.

Table 6-23 - Comparison of our recognition results with Çömez [4]

Research	CWR	WAR
Susman, 2012	78.68%	74.89%
Çömez, 2003	67.86%	52.30%

We can compare our results with [4] and [6] since the acoustic data utilized in these researches are not tremendously different than the amount of acoustic data utilized in this thesis. However, it is important to note that, our training data is nearly half the amount of acoustic data utilized in those researches. The most probable reason for our better performance is the use of 3-grams instead of bigrams in our language model. Also, it is reported in [4] that no smoothing method was employed.

In Table 6-24, we compare our research with Çömez [4] and Arısoy [2] in terms of several parameters which are significant for ASR systems.

Table 6-24 - Configuration comparison with Çömez [4] and Arısoy [2]

Research	Audio Corpus	Base Unit	N-gram	Smoothing
Çömez, 2003	7650 utterances	Word (stem-only)	2	None
		Stem/Ending	2	
Arısoy, 2009	194 hours	Word	3	Kneser-Ney
		Stem/Ending	4	
		Morph	4	
Susman, 2012	4769 utterances (5.2 hours)	Word (with stem interpolation)	3	Kneser-Ney
		Stem/Ending	4	
		Hybrid	4	
		Morph	4	

CHAPTER 7

CONCLUSION

In this research, we applied several methods in order to address the challenges of LVCSR (Large Vocabulary Continuous Speech Recognition) in Turkish. We proposed various data selection approaches in order to improve the robustness of the acoustic model when limited audio corpus is available. We also experimented with different recognition units and language models.

By utilizing data selection approaches, we attempted to overcome the insufficiency of the audio corpus to a certain extent. We obtained a 1.77% decrease in WER by applying a WER-based data selection strategy. We also experimented with empirical and perplexity based data selection approaches. With the empirical approach, we obtained a 0.27% decrease in WER. However, we could not observe any improvements in the recognition results by using a perplexity based data selection method. Copying the utterances with poor recognition rates in the audio training data proved to be a method which can be used to improve the robustness of the acoustic model.

We utilized different text corpora with different sizes in order to show the effect of the text corpus size on building n-gram language models. As expected, the greater in size the text corpus is, the better the n-gram models contributed to the recognition performance. We applied several approaches in order to extend the standard language modeling techniques. Our enhanced word model is based on interpolating the word based model with a stem-only based language model. In our experiments, word-based models have outperformed sub-word based models, which is contrary to the expected. We deem the reason of such contradiction as the insufficiency of the audio corpus utilized in this research. We observed a direct correlation between the number of unique words and the WER values. As mentioned previously, in other Turkish LVCSR theses which utilized similar amounts of acoustic data, the best recognition performance was retrieved by using word-based models, as in our case. We also experimented with a hybrid language model which is based on words, stems and endings. The recognition performance obtained by using this model is slightly better than the stem-ending based language model.

After having reached a baseline performance, we also investigated the optimal decoding parameters for the Julius decoder. By experimenting with beam-width and language model penalty parameters, we obtained a significant increase in the recognition performance.

During this research, we could not obtain a rich, Turkish audio corpus to be utilized in the domain of Turkish LVCSR. We utilized METU Turkish Microphone Speech Corpus, which is publicly available via LDC (Linguistic Data Consortium). Besides, we had difficulties in comparing the results of our research with previous works since the content of the audio material utilized in those researches are not publicly available. Our recognition results can be compared with future studies which will make use of the METU Turkish Microphone Speech Corpus.

We deem the recognition results obtained in our research to be significant in the context of bootstrapping ASR systems. In the future, we would like to repeat our experiments and elaborate on further methodologies if sufficient audio corpus becomes available.

REFERENCES

1) Hetherington, I. L., “A Characterization of the Problem of New, Out-of-Vocabulary Words in Continuous-Speech Recognition”, 1995

2) E. Arısoy, “Statistical And Discriminative Language Modeling for Turkish Large Vocabulary Continuous Speech Recognition”, 2009

3) H. Erdoğan, , O. Büyük, K. Oflazer, “Incorporating Language Constrains in Sub-Word Based Speech Recognition”, 2005

4) M. Çömez, T. Çiloğlu, “Large Vocabulary Continuous Speech Recognition for Turkish using HTK”, 2003

5) E. Mengüşoğlu and O. Deroo, “Turkish LVCSR: Database Preparation and Language Modeling for an Agglutinative Language”, 2001

6) E. Arısoy, “Turkish Dictation System For Radiology and Broadcast News Applications”, 2002

7) H. Dutağacı, “Statistical Language Models for Large Vocabulary Continuous Speech Recognition”, 2002

8) D. Jurafsky and J. H. Martin, “An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition”, Prentice Hall, 2000

9) L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, 1989

10) O. W. Kwon and J. Park, “Korean Large Vocabulary Continuous Speech Recognition with Morpheme-based Recognition Units”, 2002

11) P. Geutner, “Using Morphology Towards Better Large Vocabulary Speech Recognition Systems”, 1996

12) T. Rotovnik, M. S. Maucec and Z. Kacic, “Large Vocabulary Continuous Speech Recognition of an Inflected Language Using Stems and Endings”, Speech Communication, Vol. 49, No. 6, pp. 437 – 452, June 2007

13) D. Hakkani-Tür, K. Oflazer and G. Tür, Statistical Morphological Disambiguation for Agglutinative Languages, 2000

14) Sethu Vijayakumar and Hidemitsu Ogawa. 1999. “Improving Generalization Ability through Active Learning”. IEICE Transactions on Information and Systems, 82:480–487.

15) C. Pelaez-Moreno, Q. Zhu, B. Chen and N. Morgan. 2005. “Improving Generalization Ability through Active Learning”. In Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech 2005), pages 229–232, Lisboa, Portugal.

16) T. M. Kamm and G. G. L. Meyer. 2001. “Automatic Selection of Transcribed Training Material”. In Proc. IEEE Workshop Automatic Speech Recognition and Understanding, Lisboa, Portugal

17) T. M. Kamm and G. G. L. Meyer. 2002. “Selective sampling of training data for speech recognition”. In Proc.Human Language Technology, San Diego, USA.

18) A. B. Nagorski, L. Boves, and H. Steeneken. 2003. “In Search of Optimal Data Selection for Training of Automatic Speech Recognition Systems”. In Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, St. Thomas, US Virgin Islands.

19) Abhinav Sethy, Panayiotis G. Georgiou, Bhuvana Ramabhadran and Shrikanth S. Narayanan. 2009. “An Iterative Relative Entropy Minimization-Based Data Selection Approach for n-Gram Model Adaptation”. IEEE Transactions on Audio, Speech & Language Processing, 17:13–23.

20) E. Bocchieri, M. Riley, and M. Saraclar, “Methods for task adaptation of acoustic models with limited transcribed in-domain data”, In Proceedings of the International Conference on Spoken Language Processing (ICSLP), Jeju Island, South Korea, 2004

21) Ö. Salor, B. L. Pellom, Tolga Çiloglu, Mübeccel Demirekler, “Turkish speech corpora and recognition tools developed by porting SONIC: Towards multilingual speech recognition”, 2007

22) Haşim Sak, Tunga Güngör and Murat Saraçlar. 2008. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In GoTAL 2008, volume 5221 of LNCS, pages 417–427. Springer.

23) Mathias Creutz and Krista Lagus. 2005. Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0. Publications in Computer and Information Science Report A81, Helsinki University of Technology, March

24) S. Young, D. Ollason, V. Valtchev, P. Woodland, “The HTK Book (for HTK Version 3.4)”, Cambridge Research Laboratory, 2009

25) Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at Sixteen: Update and Outlook. In ASRU 2011.

26) A. Lee, T. Kawahara and K. Shikano. "Julius --- an open source real-time large vocabulary recognition engine." In Proc. European Conference on Speech Communication and Technology (EUROSPEECH), pp. 1691--1694, 2001.

27) Türk Dil Kurumu Ana Sayfası, http://www.tdk.gov.tr/, Last access date is 14.01.2012

28) Zemberek NLP, http://zembereknlp.blogspot.com/2006/11/kelime-istatistikleri.html, Last access date is 15.02.2012

29) E. Arısoy, M. Saraçlar, “Language Modelling Approaches for Turkish Large Vocabulary Continuous Speech Recognition Based On Lattice Rescoring”, 2006

30) X. Liu, M. J. F. Gales & P. C. Woodland, 2009, “Use of Contexts in Language Model Interpolation and Adaptation”, Cambridge University

31) H. Sak, M. Saraçlar, Tunga Güngör, “Morphology-based and Sub-word Language Modeling For Turkish Speech Recognition”

32) V. Siivola, M. Kurimo and K. Lagus, “Large Vocabulary Statistical Language Modeling for Continuous Speech Recognition in Finnish”, 2001

33) K. Oflazer, “Two-level Description of Turkish Morphology”, Literary and Linguistic Computing, 9(2):137-148, 1994

34) E. Erguvanlı, “The Function of Word Order in Turkish Grammar”, 1979

35) O. Büyük, “Sub-word Language Modeling For Turkish Speech Recognition”, 2005. Ö. Salor, B. L. Pellom, Tolga Çiloglu, Mübeccel Demirekler, “Turkish speech corpora and recognition tools developed by porting SONIC: Towards multilingual speech recognition”, 2007

36) Bo-June Hsu. 2007. Generalized Linear Interpolation of Language Models. Automatic Speech Recognition & Understanding, 2007. ASRU. MIT, Cambridge

37) Google Voice Search, http://www.google.com/insidesearch/voicesearch-chrome.html, Last access date is 07.03.2012

38) Paul Heisterkamp. Linguatronic Product-Level Speech System for Mercedes-Benz Cars. Proceedings of the first international conference on Human language technology research. Ulm, Germany

39) Home page of Blinkx, http://www.blinkx.com/, Last access date is 07.03.2012

40) Apple Siri, http://www.apple.com/iphone/features/siri.html, Last access date is 07.03.2012

41) Dikte ASR system, http://www.dikte.com.tr/, Last access date is 07.03.2012

APPENDIX A

LIST OF MERGED WORDS

bir şey, bir şeyi, bir şeye, bir şeyler, bir kaç, bir kaçı, bir kaçını, hiç bir, hiç biri, ya da, hem de, bir çok, bir çoğu, her iki, her ikisi, bir kez, bir an, her zaman, tabii ki, en çok, her hangi, pek çok, belki de, her bir, her biri, en fazla, en az, pek de, hiç de, çok az, hiç kimse, hiç birşey, hiç birşeyi, hiç birşeye

APPENDIX B

DEFINITIONS OF OPTIMIZED DECODING PARAMETERS

Beam Width

Beam width in number of HMM nodes for rank beaming. This value defines search width on the 1st pass, and has dominant effect on the total processing time. Smaller width will speed up the decoding, but too small value will result in a substantial increase of recognition errors due to search failure. Larger value will make the search stable and will lead to failure-free search, but processing time will grow in proportion to the width.

Language Model Penalty

(N-gram) Language model weights and word insertion penalties.