STATISTICAL AND DISCRIMINATIVE LANGUAGE MODELING FOR TURKISH LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

by

Ebru Arısoy

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2002

M.S., Electrical and Electronics Engineering, Boğaziçi University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2009

STATISTICAL AND DISCRIMINATIVE LANGUAGE MODELING FOR TURKISH LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

APPROVED BY:

Asst. Prof. Murat Saraçlar (Thesis Supervisor)	.....
Prof. Ethem Alpaydın	.....
Prof. Levent M. Arslan	.....
Asst. Prof. Hakan Erdoğan	.....
Assoc. Prof. Mikko Kurimo	.....
Prof. Bülent Sankur	.....

DATE OF APPROVAL: 17.12.2009

to my parents

ACKNOWLEDGEMENTS

I am very grateful to my thesis supervisor Murat Saraçlar for his invaluable guidance, brilliant ideas, infinite encouragement and patience. Meeting with him has completely changed my life and academic career. He is an excellent advisor and I am fortunate to be one of his students.

I would like to thank Hakan Erdoğan, Ethem Alpaydın and Levent Arslan for being members of the thesis follow-up committee and defense jury. Their precious feedbacks contributed a lot to the work in this dissertation. I would like to thank Mikko Kurimo for travelling from Helsinki for participating in my jury and for his valuable feedbacks during my defense. I owe special thanks to Bülent Sankur for his support and encouragement for the last seven years of my graduate study and for being a member of my jury. Feeling his support is extremely valuable to me.

During my Ph.D., I studied abroad for 17 months as a visiting research scholar. I would like to thank Brian Roark and Izhak Shafran for giving me the opportunity to work with them at Oregon Graduate Institute (OGI), Richard Sproat for giving me the opportunity to work with him at University of Illinois at Urbana-Champaign (UIUC) and Mikko Kurimo for giving me the opportunity to work with him and his colleagues at Helsinki University of Technology (HUT). I have special thanks to Brian for his brilliant ideas and guidance with his joyful attitude and to Izhak, Richard and Mikko for their invaluable guidance and help. Also, I would like to thank Bhuvana Ramabhadran and Ruhi Sarıkaya for giving me the opportunity visit an industry lab for two weeks. My special thanks go to Ruhi for hosting me in his house and making all the arrangements for my visit.

Additionally, my thanks go to Haşim Sak, Doğan Can, Sıddıka Parlak, Thomas Pellegrini, Mathias Creutz and Teemu Hirsimäki for their scientific contributions to this dissertation.

Many thanks to the following people and the institutions for providing me with their data and software tools: Haşim Sak for morphological analyzer, morphological disambiguation tool and text corpus; Gülşen Eryiğit for dependency analyzer; Tolga Çiloğlu, Hakan Erdoğan and Helin Dutağacı for acoustic and text corpora; AT&T – Labs Research for the software. My special thanks go to Kemal Oflazer for providing me with morphological analyses of Turkish words, always earlier than I asked for.

I would like to thank my friends at BÜSİM for making the lab a warm and fun environment. Many thanks to Ceyhun Akgül, Sergül Aydöre, Doğaç Başaran, Doğan Can, Oya Çeliktutan, Hatice Çınar, Cem Demirkır, Çağlayan Dicle, Erinç Dikici, Helin Dutağacı, Bilgin Eşme, Neslihan Gerek, Sıddıka Parlak, Arman Savran, Temuçin Som, Ekin Şahin, İpek Şen and Sinan Yıldırım. Special thanks go to Oya and Gökhan for keeping me awake in the lab after midnight in the last months of this dissertation.

I would like to thank my friends in the "day of gold" members, Esra, Zeynep, Serda, Suncem, Ender, Çiçek and Işılay, for their patience and support during my long years of research. I owe special thanks to Suna Eryiğit for her endless support and friendship. Thanks Suna for being there as my best friend.

Finally, I would like to express my gratitude to my parents and to my sister and brother who have supported me with endless love and understanding throughout my life. This dissertation is dedicated to them. My sweet cat Meze Kepçe also deserves my thanks for the enjoyment she gave.

This thesis was supported in part by the Scientific and Technical Research Council of Turkey (TÜBİTAK) Integrated Doctorate Program (BDP), in part by TÜBİTAK Career Project (105E102), in part by Boğaziçi University Research Fund (BAP) Projects (05HA202 and 07HA201D), in part by Turkish State Planning Organization (DPT) under the project number DPT2007K120610. This thesis was also supported by Turkish Academy of Sciences (TÜBA) as a part of Murat Saraçlar's TÜBA-GEBIP award. My visits to OGI and UIUC were supported by TÜBİTAK-BDP and to HUT was partially supported by SIMILAR Network of Excellence within EU's 6th Framework Program.

ABSTRACT

STATISTICAL AND DISCRIMINATIVE LANGUAGE MODELING FOR TURKISH LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

Turkish, being an agglutinative language with rich morphology, presents challenges for Large Vocabulary Continuous Speech Recognition (LVCSR) systems. First, the agglutinative nature of Turkish leads to a high number of Out-of-Vocabulary (OOV) words which in turn lower Automatic Speech Recognition (ASR) accuracy. Second, Turkish has a relatively free word order that leads to non-robust language model estimates. These challenges have been mostly handled by using meaningful segmentations of words, called sub-lexical units, in language modeling. However, a shortcoming of sub-lexical units is over-generation which needs to be dealt with for higher accuracies.

This dissertation aims to address the challenges of Turkish in LVCSR. Grammatical and statistical sub-lexical units for language modeling are investigated and they yield substantial improvements over the word language models. Our novel approach inspired by dynamic vocabulary adaptation mostly recovers the errors caused by over-generation and further improves the accuracy of sub-lexical units. Additionally, discriminative language models (DLMs) with linguistically and statistically motivated features are utilized. DLM outperforms the conventional approaches, partly due to the improved parameter estimates with discriminative training and partly due to integrating the complex language characteristics of Turkish into language modeling.

The significance of this dissertation lies in being a comparative study of several sub-lexical units on the same LVCSR system, addressing the over-generation problem of sub-lexical units and extending sub-lexical-based generative language modeling of Turkish to discriminative language modeling. These approaches can be easily extended to other morphologically rich languages that suffer from similar problems.

ÖZET

TÜRKÇE GENİŞ DAĞARCIKLI KONUŞMA TANIMA İÇİN İSTATİSTİKSEL VE AYIRICI DİL MODELLEMESİ

Sondan eklemeli ve zengin biçimbilimsel yapıya sahip olan Türkçe, Geniş Dağarcıklı Sürekli Konuşma Tanıma (GDSKT) için zor bir dildir. Türkçe'nin sondan eklemeli yapısı yüzünden çok fazla sayıda dağarcık dışı kelime bulunmakta ve bu kelimelerin varlığı konuşma tanıma başarımlarını düşürmektedir. Türkçe'nin göreceli serbest kelime dizilimine sahip olması gürbüz olmayan dil modeli kestirimlerine sebep olmaktadır. Bu zorluklar, kelime-altı birimler olarak adlandırılan, anlamlı kelime parçacıklarının dil modellemesinde kullanılmasıyla büyük ölçüde aşılmıştır. Fakat, kelime-altı birimlerin bir kusuru Türkçe'de bulunmayan kelimeleri fazladan üretmesidir. Daha yüksek başarımlar için bu sorunun da çözülmesi gerekmektedir.

Bu tez Türkçe'nin GDSKT sistemlerindeki zorluklarını çözmeyi hedeflemektedir. Dil modellemesi için dilbilimsel ve istatistiksel kelime-altı birimler araştırılmış ve bu birimler kelime dil modelleri üzerinden anlamlı başarım artışı sağlamıştır. Dinamik dağarcık uyarlamasından esinlenerek önerdiğimiz yeni yaklaşımımız kelime-altı birimlerdeki fazladan üretilen kelime hatalarını büyük ölçüde düzeltmiş ve kelime-altı birimlerin başarımını daha da arttırmıştır. Ayrıca Ayırıcı Dil Modelleri (ADM) dilbilimsel ve istatistiksel özniteliklerle birlikte kullanılmıştır. Hem ayırıcı eğitim ile daha iyi parametre kestirimleri sağlaması, hem de Türkçe'nin dil özelliklerini dil modellemesine katması sayesinde ADM geleneksel yöntemlerden daha iyi başarımlar göstermiştir.

Bu tezin önemi birçok kelime-altı birimi aynı GDSKT sisteminde karşılaştıran bir çalışma olmasında, kelime-altı birimlerdeki fazladan kelime üretme hatalarını düzeltmeye çalışmasında ve Türkçe dil modelleme yaklaşımlarına ADM'ni eklemesinde yatmaktadır. Önerilen yöntemler benzer sorunları yaşayan diğer zengin biçimbilimsel yapıya sahip dillere kolaylıkla genişletilebilir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

a	Acoustic feature vector
A	Acoustic observations (sequence of acoustic feature vectors)
$\mathcal{A}$	Alphabet for acoustic feature vectors
$\mathcal{C}$	Set of clusters
$c(\cdot, \cdot)$	Cost between two letters
c_k	Class of the k'th word w_k
$C(w_{k-n+1} \ \ldots \ w_k)$	Number of occurrences of a word string
$d(\cdot, \cdot)$	Distance between two strings/clusters
$D(\cdot, \cdot)$	Pairwise string distance function
h	History
H_0	Null hypothesis
H_1	Alternative hypothesis
f_k^i	i'th factor of the k'th word w_k
$f(h, w)$	An arbitrary feature as a function of h and w
$f(W)$	An arbitrary feature as a function of W
$GEN(x)$	A function enumerating a finite set of candidates for inputs
$H(\cdot)$	Entropy
$I(\cdot, \cdot)$	Average mutual information
$L(\cdot)$	Log-likelihood
$P(\cdot)$	Prior probability
$P(\cdot\|\cdot)$	Conditional probability
$P(\cdot, \cdot)$	Joint probability
$r(\cdot)$	Root/First-morph function
t_k	Tag of the k'th word w_k
$\mathcal{V}$	Recognition vocabulary
w	word
W	Word string
(x, y)	Input/Output pairs
$\mathcal{X}$	Set of all possible inputs

$\mathcal{Y}$	Set of all possible outputs
Z	Global normalization constant
$Z(\cdot)$	Normalization constant as a function of some parameters
$\bar{\alpha}$	Parameter Vector
β	Language model weight
Δ	Delta coefficients
$\Delta\Delta$	Delta-delta coefficients
λ	Model parameters
$\Phi(w_1 \ \ldots \ w_{k-1})$	Equivalence class for the word string
$\Phi(x, y)$	Real-valued feature vector
$\Phi_i(x, y)$	i'th element of the feature vector
$\pi(\cdot)$	Class assignment function
τ	threshold
ADM	Ayırıcı Dil Modelleri
ASR	Automatic Speech Recognition
AUL	Average Unit Length
BAP	Boğaziçi University Research Fund
BN	Broadcast News
CDG	Constraint Dependency Grammar
CRF	Conditional Random Field
DARPA	The Defense Advanced Research Projects Agency
DF	Distinctive Feature
DLM	Discriminative Language Model
DPT	Turkish State Planning Organization
EARS	Effective, Affordable, Reusable Speech-to-Text
FLM	Factored Language Model
FST	Finite State Transducer
GDSKT	Geniş Dağarcıklı Sürekli Konuşma Tanıma
GMM	Gaussian Mixture Model

H2H	Head-to-Head
HDLA	Hypothesis Driven Lexical Adaptation
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
HUT	Helsinki University of Technology
IG	Inflectional Group
IM	Initial Morph
IV	In-Vocabulary
LDC	Linguistic Data Consortium
LM	Language Model
LVCSR	Large Vocabulary Continuous Speech Recognition
LWER	Lattice Word Error Rate
MAPSSWE	Matched Pair Sentence Segment Word Error
MDL	Minimum Description Length
ME	Maximum Entropy
MED	Minimum Edit Distance
METU	Middle East Technical University
MFCC	Mel Frequency Cepstral Coefficients
MLE	Maximum Likelihood Estimation
MMI	Maximum Mutual Information
MPE	Minimum Phone Error
NIM	Non-initial Morph
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
OGI	Oregon Graduate Institute
OOV	Out-of-Vocabulary
PCM	Pulse Code Modulation
PDMED	Position Dependent Minimum Edit Distance
PoS	Part-of-Speech
RS	Read Speech
RTF	Real Time Factor

SOV	Subject-Object-Verb
SRILM	SRI Language Modeling
STM	Segment Time Marked
SVM	Support Vector Machine
TÜBİTAK	The Scientific and Technological Research Council of Turkey
UIUC	University of Illinois at Urbana-Champaign
UPW	Units Per Word
VOS	Verb-Object-Subject
WAV	Waveform Audio Format
WB	Word Boundary
WER	Word Error Rate

1. INTRODUCTION

This dissertation explores statistical and discriminative language modeling for Turkish LVCSR. A statistical language model gives a probability distribution over all possible word strings in a language. The ultimate goal in statistical language modeling is to find probability estimates for the word strings that are as close as possible to their true distribution. Therefore, several statistical techniques have been proposed to appropriately model natural languages. These techniques employ large amounts of text data to robustly estimate the model parameters. Statistical language models are used in many natural language technologies, such as speech recognition, machine translation, handwriting recognition, spelling correction, as the crucial component to improve the system performance. Therefore, the quality of the statistical language models can be directly evaluated on the system performance. An alternative approach, without including the system into the evaluation, is to measure the generalization capacity of the proposed language model on a separate text that is not seen during model training. In that case, the average log-likelihood of the test text is used to assess the quality of the language model. However, this approach is not directly correlated with the system performance.

In the state-of-the-art, n-grams are the conventional language modeling approach due to their simplicity with their substantial modeling performance. In n-grams, the probability of a word string is approximated with the product of conditional word probabilities, conditioned on n-1 previous words. The conditional probabilities are estimated using a large text corpus with Maximum Likelihood Estimation (MLE). Discriminative training of language models have been recently introduced to obtain improved parameter estimates for language models. In contrast to conventional n-gram language models, discriminative language modelling is a feature-based approach and the parameters are estimated with discriminative training algorithms. Discriminative language models have been demonstrated to consistently outperform generative language modeling approaches partly due to the improved parameter estimation with discriminative training and partly due to the ease with which many overlapping rele-

vant features can be integrated into language modeling.

In this thesis, we only focus on statistical and dicriminative language modeling for ASR, especially for LVCSR which is developed for transcribing large vocabulary continuous speech, as opposed to small vocabulary or isolated speech.

1.1. Statement of the Problem

Turkish, being an agglutinative language with rich morphology, presents challenges for LVCSR systems. This thesis is an attempt to handle the challenges of Turkish in speech recognition. This section states the problems that are encountered with the state-of-the-art language modeling approaches. The next section will present the contributions of this dissertation to language modeling in LVCSR of Turkish, as well as other agglutinative or highly inflectional languages.

The challenges of Turkish introduced to LVCSR systems are (i) high OOV rates, (ii) non-robust n-gram estimates, (iii) ungrammatical recognition outputs when sub-lexical units are used as language modeling units. Sub-lexical units are meaningful word segments, e.g., morphemes of a language, and the language models trained on these segments may cause invalid sub-lexical sequences in speech recognition. The following paragraphs will explain the challenges of Turkish in detail.

The state-of-the-art LVCSR systems utilize predetermined and finite recognition vocabularies that contain the most frequent words related to the speech recognition domain. The words that do not occur in the ASR vocabulary but that are uttered by the speaker are called OOV words. Existence of OOV words is a significant source of recognition errors in ASR since if a word is not in the vocabulary, it has no chance to be recognized correctly. Unfortunately, the productive morphology of Turkish yields many unique word forms, making it difficult to have a vocabulary covering all these words. Therefore, the number of OOV words are quite high for Turkish even for vocabulary sizes that would be considered as large for English.

Another problem introduced to ASR systems by the language characteristics of Turkish is the non-robust n-gram estimates. Turkish has a relatively free word order where the order of constituents can be changed without affecting the grammaticality of a sentence. This relatively free word order causes sparse text data and sparse data leads to non-robust n-gram language model estimates. As an easy way of handling the OOV problem, increasing the number of vocabulary words can be proposed. However, in addition to the sparseness due to the relatively free word order, large vocabularies also make the robust estimation of n-gram parameters even more difficult. ASR systems with large vocabularies need a huge amount of text for robust language model estimates since there are millions of parameters that need to be estimated in large vocabulary systems. Additionally, large vocabulary ASR systems require more memory and computational power and it may not be possible to accommodate very large vocabularies and training data due to computational limitations.

Other agglutinative languages also suffer from high OOV rates and utilizing vocabularies composed of sub-lexical recognition units have been proposed for these languages, as well as for Turkish. Sub-lexical recognition units are meaningful word segments that are obtained with rule-based or statistical approaches. When using sub-lexical vocabularies, the language models are built in the same way with word vocabularies by considering sub-lexical units as if they are words. It has been shown that sub-lexical vocabularies handle the OOV problem with moderate vocabulary sizes and result in more robust n-gram estimates. In Turkish, phonological processes determine the harmony of the sounds within words and between morpheme boundaries in suffixation and morphephonemic alternations determine some of the sound changes in suffixation. In other words, same stems and suffixes in lexical form may correspond to different stems and suffixes in surface form due to phonological and morphophonemic properties of Turkish. These properties contribute to the vocabulary growth problem in Turkish, consequently, OOV words are introduced to ASR systems even with very large word vocabularies. In terms of sub-lexical recognition vocabularies, these properties are the major issues that need to be taken into account during or after speech recognition. Sub-lexical recognition outputs are concatenated in order to obtain the words. The generated words should obey the phonological and morphophonemic rules

of Turkish. However the language model itself does not guarantee that these rules will be satisfied. Therefore, the proposed sub-lexical approaches introduce their own problem of generating invalid word sequences while addressing the OOV problem.

1.2. Main Contributions of the Thesis

ASR research for Turkish has grown rapidly in the last 10 years. Existence of shared language resources is very crucial in the ASR research for system training and development. However, there has been no publicly available language resources for Turkish until recently. Therefore, most of the researchers working on Turkish ASR have collected their own data for system training. The general approach employed in Turkish ASR research is to investigate the effectiveness of different sub-lexical recognition units. Due to the lack of standard and publicly available language resources, the results of the previous studies on Turkish ASR are not comparable with each other.

In this thesis, we aim to investigate novel language modeling approaches to address the challenges of Turkish in LVCSR. To our knowledge, our work is the most comprehensive research on Turkish language modeling. First, we compare previously proposed sub-lexical language modeling approaches as well as our novel approaches in the same LVCSR system trained on large databases. Second, our research extends Turkish language modeling research framework from sub-lexical-based generative modeling to discriminative modeling with linguistically and statistically motivated features. The main contributions of the thesis together with the main scientific publications are listed as follows:

1. Previously proposed sub-lexical language modeling approaches are compared with each other and with very large vocabulary word language models in the same LVCSR system trained on large acoustic and text databases. This provides comparative baseline results both for word and sub-lexical language modeling studies for Turkish. Sub-lexical units perform even better than our largest word vocabulary yielding only 1.0 per cent OOV rate. This part of the thesis was published in the broadcast news transcription section of [1].

2. Lexical form sub-lexical language modeling units[1] are proposed both to obtain more robust n-gram estimates with sub-lexical units and to deal with the ungrammatical word problem to an extent directly during speech recognition. In this approach phonological and morphophonemic variations of the same morpheme is introduced as a single unit in language modeling. Lexical form sub-lexical approach only guarantees the correct morphophonemics, not the correct morphotactics. Lexical form sub-lexical units have been shown to outperform their surface form counterparts. This part of the thesis was published in [2].

3. We perform a manual error analysis on the recognition outputs of the best scoring baseline ASR system to understand the sources of the recognition errors. Our analysis reveals important results that lead us to investigate corrective language models for Turkish. This part of the thesis was published in [3].

4. Dynamic vocabulary adaptation approaches that have already been proposed for highly inflected languages are also utilized in Turkish LVCSR to handle the OOV problem at word vocabulary level. Here, our novel contribution is that dynamic vocabulary adaptation techniques are modified to deal with the over-generation problem in sub-lexical language modeling units. Even though our approach brings back the OOV problem, significant improvements are achieved on the baseline sub-lexical system. This part of the thesis was published in [4].

5. We utilize discriminative language models as a complementary approach to conventional n-grams to correct recognition errors in the baseline ASR outputs. Discriminative language modeling provides better recognition performance due to discriminatively trained parameter estimates and due to integrating the challenging structure of Turkish into language modeling via overlapping features. Here the novelty of our research is in the linguistically and statistically motivated features proposed for Turkish. The work with morphology-based features was published in [5] and the work with syntactic-based and statistical-based features will be

[1] This is a joint work with Haşim Sak.

published in [6].

6. Discriminative language models are also investigated on the ASR output of the sub-lexical recognition units to explore the effect of recognition units on discriminative language modeling. Sub-lexical recognition units perform better than words also in discriminative language models. This finding was published in [1].

Even though the language modeling approaches in this thesis are proposed for Turkish, some of our approaches are language independent. Therefore they can be easily extended to other agglutinative or highly inflectional languages that suffer from similar problems in ASR systems. We believe that our language dependent approaches can also help the speech recognition systems of these languages to further improve the currently possible accuracies after simple modifications.

1.3. Organization of The Thesis

This dissertation is organized as follows: In Chapter 2 we give an overview of the main components of ASR systems with an emphasis on statistical language models, an introduction to Turkish language with its challenges for ASR systems and a review of the previous work that this dissertation is based on. Chapter 3 explains the language resources utilized in this thesis and the details of the Turkish LVCSR systems developed in this thesis for investigating our proposed language modeling approaches.

The novel contributions of this research to Turkish ASR are explained in Chapters 4, 5 and 6. We explore grammatical and statistical sub-lexical recognition units and compare their performances on the same LVCSR system in Chapter 4. The findings of the manual error analysis are also reported in this Chapter. Chapter 5 investigates dynamic vocabulary adaptation techniques on Turkish and their extension for solving ungrammatical word problem in sub-lexical recognition outputs. Discriminative language modeling for Turkish is explained in Chapter 6 with an emphasis on the proposed linguistically and statistically motivated feature sets. Finally Chapter 7 concludes this dissertation with a summary of our findings and the future directions for our research.

2. BACKGROUND

The main focus of this dissertation is investigating novel language modeling approaches for LVCSR of Turkish. This chapter gives a background about the main components of ASR systems with an emphasis on language modeling, about Turkish language with its challenges for ASR systems and about the previous work that this dissertation is based on.

2.1. Foundations of Automatic Speech Recognition

The aim of ASR is to automatically produce the transcription of the input speech. There are various ASR applications, such as digit recognition, recognition of yes/no answers to questions, dictation, transcription of telephone conversations or broadcast news, etc. The difficulty of these applications depends on the vocabulary size (small or large), speaking mode (isolated or continuous), speaking style (planned or spontaneous speech) and acoustic conditions (clean, noisy conditions or telephone speech) [7]. For instance large vocabulary, continuous and spontaneous telephone speech recognition under noisy acoustic conditions is a much more difficult task than small vocabulary isolated word recognition with clean speech. The target ASR applications in this thesis are LVCSR tasks which are also challenging applications.

ASR is also used as one of the main components in spoken document retrieval systems which aim to retrieve audio clips related to a query, in spoken term detection systems which aim to locate occurrences of a term in a spoken archive, and in spoken dialogue systems which aim to accomplish a task given with spoken language. Additionally, speech-to-speech translation systems also require ASR technology to obtain accurate transcriptions for the input to the machine translation component preceding the text-to-speech synthesis component.

2.1.1. Components of an ASR System

Speech recognition problem can be explained with a source-channel model as given in Figure 2.1, taken from [8]. Here, the source is the speaker's mind. The speaker decides on which word string he will utter using his huge vocabulary and his linguistic knowledge. The domain of the (ASR) application also determines the speaker's choices of words from his vocabulary. For instance the speakers' word choices for a dictation system for a medical domain is completely different than the speakers' word choices for a spoken dialog system developed for buying train tickets. Then, this word string passes through a noisy acoustic channel which is composed of the speech producer, the transmission channel and the acoustic processor (front-end) parts. Speech is produced by the speaker and converted into the acoustic observations, A, by the front-end. Linguistic decoder picks the most likely word string corresponding to the speech input using the statistical acoustic and language models.

In Figure 2.1, W denotes a string of n words taken from the vocabulary of any language. However, for ASR purposes we assume a fixed and finite vocabulary $\mathcal{V}$, i.e., $W = w_1\ w_2\ \ldots\ w_n$ where $w_i \in \mathcal{V}$. $\mathcal{V}$ is the recognition vocabulary of the ASR system and $|\mathcal{V}|$ is the size of this vocabulary. Vocabulary size can be as small as two words for recognition of yes/no answers to questions or 10 digits for digit recognition and as large as thousands of words for dictation systems. Note that the decoder can only recognize the words in its vocabulary. Therefore, the vocabulary of the ASR system has to match with speaker's possible word choices. If the speaker utters a word which is not in the recognition vocabulary, this word is called an OOV word and it will be

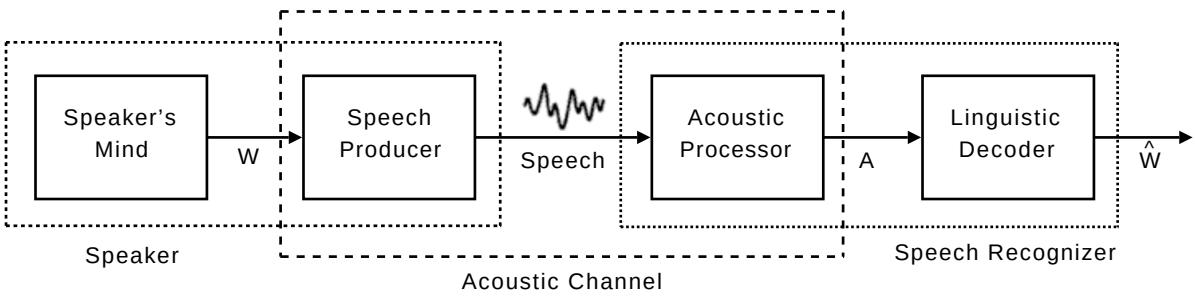


Figure 2.1. Source-channel model of speech recognition [8].

exchanged with one of the words in the recognition vocabulary in $\hat{W}$.

In Figure 2.1, A is the acoustic evidence that contains a sequence of m symbols taken from an alphabet $\mathcal{A}$, i.e., $A = a_1\ a_2\ \ldots\ a_m$ where $a_i \in \mathcal{A}$. A is obtained from the speech signal with acoustic processing. **Acoustic processor (front-end)** is one of the main components of an ASR system and it is required to convert the speech signal into the form that can be processed by digital computers. In this thesis, we utilize Mel Frequency Cepstral Coefficients (MFCC) together with Δ and $\Delta\Delta$ coefficients as the features. So, a_i corresponds to a 39 dimensional feature vector containing MFCC, Δ and $\Delta\Delta$ coefficients obtained from the speech signal. The details of front-end design can be found in [7, 9].

ASR problem, stated as finding the most probable word string among all possible word strings when the input speech is given, is formulated as follows:

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W|A) \tag{2.1}$$

Using the Bayes' formula, Equation 2.1 can be rewritten as

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, \frac{P(A|W)P(W)}{P(A)} \tag{2.2}$$

where $P(A|W)$ is the probability of observing the acoustic evidence A when the speaker utters the word string W and $P(W)$ is the probability of saying the word string W. $P(A)$ represents the probability of observing A, however, there is no need to use this probability in finding the word string maximizing the Equation 2.2. So Equation 2.2 simplifies to the below form

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(A|W)P(W) \tag{2.3}$$

which is "the fundamental equation of speech recognition". In order to find the most likely word sequence, the decoder needs to know the probabilities given in Equation 2.3. Statistical acoustic and language models are the two essential components in this equa-

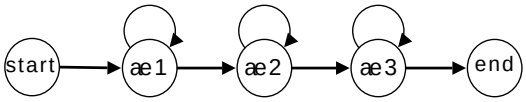


Figure 2.2. 3-state HMM model for phone "æ".

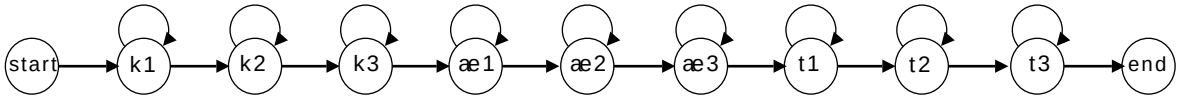


Figure 2.3. A composite HMM model for the word "cat" pronounced as "k æ t".

tion. Acoustic model is responsible for assigning probabilities to the acoustic observations when the word string is given, $P(A|W)$, and the language model is responsible for assigning probabilities to the word string, $P(W)$. In the state-of-the-art ASR systems Hidden Markov Models (HMMs) are utilized as acoustic models and n-grams are utilized as language models.

In ASR systems, a simple approach for **acoustic modeling** is to model each phone in a language using a 3-state left-to-right HMM. Figure 2.2 shows an example phone model for the phone "æ". If words are used as vocabulary items, then composite HMM models are generated for each word in the vocabulary. Figure 2.3 shows a composite HMM model for the word "cat" pronounced as "k æ t". Note that the acoustic models are built for phones, so a lexicon yielding the correct pronunciations of words are required for acoustic modeling. The HMM model parameters are learned using a large set of A and corresponding W pairs. In most of the ASR systems, triphone HMM models are utilized instead of phone models to take the right and the left context of the phones into account. For instance, the possible triphones for the word "cat" are as follows: "ϵ−k+æ", "k-æ+t", "æ-t+ϵ" where "-" sign represents the left context, "+" sign represents the right context and ϵ represents the word boundary. In order to handle the data sparsity in triphones, the HMM states are clustered. The output distribution of each HMM state is modelled with Gaussian Mixture Models (GMMs). See [8, 7, 9, 10] for the details of HMMs and acoustic models.

In order to assign probabilities to word strings, **language modeling** needs to model the speaker's knowledge of the language. n-grams are utilized as the language

model and their parameters are learned from a large text corpus related the ASR domain. This dissertation focuses on the language modeling component of the speech recognizer. Therefore, we will give a more detailed information on this component in the next section.

The **decoder** is another essential component corresponding to the $\underset{W}{\text{argmax}}$ part of Equation 2.2. The decoder implements a hypothesis search algorithm and it is responsible for finding the most likely word string among all possible word strings. If there are n words in a word string and if the ASR vocabulary is composed of $|\mathcal{V}|$ words, then there will be totally $|\mathcal{V}|^n$ possible word strings that need to be searched by the decoder to find the most probable word string. Acoustic and language models help the decoder to restrict the search space since low probability paths are pruned from the search space during decoding. Efficient decoding algorithms make the search tractable even for very large vocabulary sizes. In Equation 2.2, $\hat{W}$ corresponds to the most likely word string (1-best hypothesis) that can be obtained from the decoder with the current acoustic and language models.

The most likely hypothesis may not be the same with the word string that is uttered by the speaker. The possible causes of this include the lack of some of the speaker's words in the recognition vocabulary (OOV words), the imperfect modeling of the speaker's language knowledge with statistical language models, the imperfect acoustic models and the astronomically large number of word strings in decoder's search space. These causes result in word errors in the hypothesis strings. The ratio of the total number of errors in the hypothesis strings to the total number of words in the reference strings, called word error rate (WER), defines the performance of the speech recognizer. The number of errors in each hypothesis string is calculated with the Levenshtein minimum edit distance algorithm [11]. The WER is formulated as follows:

$$\text{WER (per cent)} = \frac{\#D + \#S + \#I}{\#\text{reference words}} \times 100 \tag{2.4}$$

Here $\#D$, $\#S$ and $\#I$ respectively represent the minimum number of deletion, sub-

stitution and insertion operations required to transform the hypothesis strings to the reference strings.

In addition to the 1-best hypothesis, a more sophisticated ASR output, lattice or N-best list, can be obtained from the decoder. A word lattice is an efficient representation of the possible word sequences in the form of a directed graph [7]. Figure 2.4 shows an example lattice output from the decoder. In this representation, the arcs are labeled with words and the associated weights. The weights can be the negative log-likelihood obtained from acoustic and language models. An alternative is to get the posterior probabilities by normalizing the weights over the lattice. In Figure 2.4, the weights are posterior probabilities. The most likely hypothesis obtained from the lattice is "`haberler sundu`" and it has an error rate of 50 per cent. However, the path with the lowest WER is "`haberleri sundu`". Oracle error rate or lattice word error rate is the error rate of the path in the lattice with the lowest word error. This is the lower bound on the error rate that can be achieved from the lattice. In this example, the oracle error rate is 0 per cent since the hypothesis with the lowest WER exactly matches with the reference string. An N-best list contains the most probable N hypotheses from the lattice output. There are six hypotheses that can be extracted from the lattice given in Figure 2.4. The 6-best list hypotheses with their probabilities are as follows:

`haberler sundu` 0.48

`haberleri sundu` 0.24

`haberler sorduk` 0.12

`hangi evi sundu` 0.08

`haberleri sorduk` 0.06

`hangi evi sorduk` 0.02

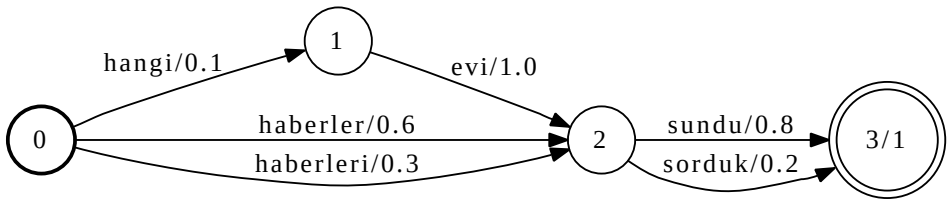


Figure 2.4. Lattice example for the reference sentence "haberleri sundu".

Note that the correct hypothesis is the second best hypothesis in the list.

2.1.2. Statistical Language Models

Statistical language models assign a prior probability, $P(W)$, to every word string that the speaker wishes to say. Assigning exact prior probabilities to word strings means perfect modeling of the language. However, it is not very realistic to achieve this with simple statistical models since the language has a very deep structure. Therefore, statistical language models try to find an appropriate estimate for $P(W)$ by capturing the regularities of the language.

If W is a string of n words, then the prior probability of this string is decomposed into the following form using the chain rule.

$$P(W) = P(w_1\ w_2\ \ldots\ w_N) = \prod_{k=1}^{N} P(w_k|w_1\ \ldots\ w_{k-1}) \tag{2.5}$$

Here the prior probability is calculated in terms of the dependencies of words to a group of preceding words, $w_1\ \ldots\ w_{k-1}$, which is called the "history". We need to estimate these conditional probabilities in order to assign a probability to $P(W)$. Due to two different reasons, it is not rational to obtain the prior probability as given in Equation 2.5. First, if the history is too long, it is not possible to robustly estimate the conditional probabilities, $P(w_k|w_1\ \ldots\ w_{k-1})$, since there are several variables. Second, it is not entirely true that the speakers' choice of a word depends on all the words in it's entire history. Therefore, it will be a more realistic assumption to put the history into an equivalence class, $\Phi(w_1\ \ldots\ w_{k-1})$, as suggested in [8]. Equivalence classes change Equation 2.5 into the following form.

$$P(W) = P(w_1\ w_2\ \ldots\ w_N) = \prod_{k=1}^{N} P(w_k|\Phi(w_1\ \ldots\ w_{k-1})) \tag{2.6}$$

Equivalence classes can be any classification of the words in the history or their syntactic and semantic information. However, the most common approach in state-

of-the art ASR technology does not take the complex structure of the language into account. It is only based on a very simple equivalence classification which utilizes only the n-1 preceding words as the history. This approach results in n-gram language models, $P(w_k|w_{k-n+1}\dots\ w_{k-1})$. Then, $P(W)$ is approximated with n-gram language models, given as follows:

$$P(W) = P(w_1\ w_2\ \dots\ w_N) \approx \prod_{k=1}^{N} P(w_k|w_{k-n+1}\dots\ w_{k-1}) \tag{2.7}$$

n-gram language model probabilities are estimated from a text corpus related to the ASR application domain with MLE. In other words, n-gram probabilities are estimated by counting the occurrences of a particular n-gram in the text data and dividing this count to the number of occurrences of all n-grams that start with the same sequence of n-1 words, given as follows:

$$P(w_k|w_{k-n+1}\ \dots\ w_{k-1}) = \frac{C(w_{k-n+1}\ \dots\ w_{k-1}\ w_k)}{C(w_{k-n+1}\ \dots\ w_{k-1})} \tag{2.8}$$

where $C(\cdot)$ represents the number of occurrences of the word string given in parentheses in the text data.

In statistical language models, large number of parameters need to be estimated with MLE and robust estimation of these parameters critically depend on the availability of large amounts of text data. See [12] for the effect of the amount of the text data in language modeling to the WER of the ASR system. If the ASR vocabulary contains $|\mathcal{V}|$ words, then there will be $|\mathcal{V}|^n$ probabilities that we need to calculate. Consequently, higher order n-grams increase the number of language model parameters exponentially. Therefore, there is always a trade-off between n-gram order and robustness of the n-gram parameters depending on the amount of the text data. As a result, 3-grams, where the history contains only the preceding two words, are the most frequently used n-gram order in ASR applications.

One of the problems in n-gram language modeling is data sparseness. If the

training corpus is not large enough, then extremely small or zero probabilities can be assigned to many possible word sequences. In order to prevent elimination of possible word sequences during decoding, generally, n-gram smoothing is applied to artificially increase small probabilities and at the same time to decrease high probabilities. As a result, smoothing techniques produce better estimates for unseen data. Interpolated and backoff smoothing are the most common smoothing methods. In interpolated smoothing higher and lower order n-gram models are linearly interpolated and in backoff smoothing we backoff to a lower order n-gram model when the current model has a zero probability. Good-Turing, Katz and Kneser-Ney are some examples of popular smoothing algorithms. See [13] for a survey of smoothing approaches for statistical language models.

n-grams are the most common language modeling approach in the state-of-the-art ASR technology. However, there are several other language modeling approaches proposed to deal with the deficiencies of the traditional n-grams. See [14] for a survey of major statistical language modeling techniques. We will utilize n-gram language models and additionally investigate discriminative language models in this thesis.

2.2. Turkish Automatic Speech Recognition

Turkish is a challenging language for ASR applications. This section will first explain the main characteristics of Turkish and then the challenges introduced to large vocabulary speech recognition systems by these characteristics.

2.2.1. Characteristics of Turkish

Turkish is a member of Altaic family of languages with 29 graphemes: 8 vowels and 21 consonants. Tables 2.1 and 2.2 give respectively the vowel and consonant inventories for Turkish. Vowels with associated distinctive features, posterior/anterior, open/close, rounded/unrounded, are given in the vowel inventories. Consonants are classified according to their place of articulation. The spelling of Turkish words are very close to their spoken representation. Therefore, for ASR purposes, we consider

Table 2.1. Turkish vowels with their [IPA] symbols.

	Not rounded		Rounded	
	Open	Close	Open	Close
Posteriors	a,[a]	ı,[ɯ]	o,[o]	u,[u]
Anteriors	e,[e]	i,[i]	ö,[ø]	ü,[y]

Table 2.2. Turkish consonants with their [IPA] symbols. The consonant ğ is ignored in the table since it is mostly used to lengthen the previous vowel.

Bilabial	b [b], p [p], m [m]
Labiodental	f [f], v [v]
Dental	d [d], t [t], s [s], z [z], n [n], l [ɬ]
Alveolars	r [r]
Alveopalatal	c [dʒ], ç [ʧ], ş [ʃ], j [ʒ], l [l]
Palatal	k [c], g [ɟ], y[j]
Velar	g [g], k [k], v [w]
Uvular	h [h]

Turkish as a phonetic language. Even though there are a few letters corresponding to two distinct sounds in Table 2.2, (e.g., l corresponds to [l] or [ɬ], v corresponds to [v] or [w], k corresponds to [k] or [c] and g corresponds to [g] or [ɟ]), each letter is considered as a single sound in acoustic modeling. All l, v, k and g letters are mapped to [l], [v], [k] and [g] sounds respectively.

The main characteristics of Turkish are agglutinative morphology and relatively free word order. These features distinguish Turkish as a challenging language for natural language processing and speech recognition applications.

Turkish has an agglutinative morphology where many new words can be derived from a single stem by addition of several suffixes. There are only a few instances of prefixation in Turkish but only the emphatic reduplication of prefix in adjectives,

such as `mas-mavi` *(very blue)*, `ter-temiz` *(very clean)*, is still used productively [15]. The suffixes in Turkish are categorized as derivational or inflectional in terms of their function. Inflectional suffixes mark number, person, and gender on nouns and tense, aspect, modality and person on verbs. They never produce new lexical items and maintain the grammatical category of the word after suffixation. In contrast to inflectional suffixes, derivational suffixes introduce new lexical items. They may change or maintain the grammatical category of the word after suffixation. For instance, a noun can be derived from a noun, `göz`*(eye)*`-lük` *(eye glasses)*, or a verb, `konuş`*(to speak)*`-macı` *(the speaker)*, and a verb can be derived from a verb, `konuş`*(to speak)*`-tur` *(to make someone speak)*, or a noun, `göz`*(eye)*`-le` *(to watch)*.

Examples below show nominal and verbal inflections for Turkish words with their English glosses. The verb of a sentence usually has a more complex morphological structure than other constituents. There is not a one to one correspondence between Turkish morphemes and English words. However, we can say that one Turkish word may correspond to a group of English words.

nominal inflection: `ev-im-de-ki-ler-den` *(among those in my house)*
verbal inflection: `yap-tır-ma-yabil-iyor-du-k` *(It was possible that we did not make someone do it)*

During agglutinative suffixation, the suffixes or the stems that they are attached may go through some phonological processes. Vowel harmony is the most important phonological characteristic of Turkish. It is stated as the compatibility of vowels within the words (internal vowel harmony) or between the morpheme boundaries (external vowel harmony). According to the external vowel harmony rule, a stem ending with a back/front vowel takes a suffix starting with a back/front vowel. If we consider these two words; `evler` (houses) and `kitaplar` (books), they are decomposed into their morphemes as `ev-ler` and `kitap-lar` respectively. Although, both of the words have the same morpheme in the lexical form, `-lAr`[2], the vowel of the plural morpheme is modified according to the last preceding vowel of the stem during the suffixation

[2] 'A' is the lexical symbol realized as /a/ or /e/ in surface form.

process. In addition to the vowel harmony, there is also a consonant harmony rule where the suffix initial stop should be agree in voicing with the last sound of the stem. For instance the ablative marker, shown as `-DAn`[3] in lexical form, results in four different surface forms, `-den, -dan, -ten, -tan`, due to consonant harmony and vowel harmony rules. Here are some examples; `ev-den` (from the house), `okul-dan` (from the school), `çiçek-ten` (from the flower) and `kitap-tan` (from the book).

In the vowel and consonant harmony rules the stems remain unchanged during suffixation. However there are some morphephonomic alternations that change the stem. These are alternation in the voicing of word final plosives ([p]→[b]: `kitap` (book)→`kitab-ım` (my book), [t]→[d]: `kanat` (wing)→`kanad-ım` (my wing), [k]→[g]: `renk` (color)→`reng-im` (my color) and [ʧ]→[dʒ]: `taç` (crown)→`tac-ım` (my crow)), k alternating with soft g ([k]→ ∅: `çiçek` (flower)→`çiçeğ-im` (my flower)) and gemination (`hat` (line)→`hatt-ım` (my line)). Morphophonemic processes can be explained with rules that account for these alternations, however, there are also several exceptions that do not obey these rules.

Turkish has a relatively free word order where the order of constituents can be changed without affecting the grammaticality of a sentence. The word order is changed to emphasize a word in a sentence and the word which will be emphasized is placed just before the verb. Below examples, taken from [16], show how the order of words is changed to emphasize a word in the same sentence. Both of the sentences are grammatically correct, since the order of subject, object and verb can be in six different types in Turkish. In spite of the relatively free word order, the most preferred type is Subject-Object-Verb (SOV) and the least preferred type is Verb-Object-Subject (VOS) both for the children and the adult speech [17].

`Ben çocuğa kitabı verdim`	*(I gave the book to the children)*
`Çocuğa kitabı ben verdim`	*(It was me who gave the child the book)*
`Ben kitabı çocuğa verdim`	*(It was the child to whom I gave the book)*

[3] 'D' is the lexical symbol realized as /d/ or /t/ in surface form.

2.2.2. Challenges of Turkish in ASR

Turkish, being an agglutinative language with rich morphology, presents a challenge for ASR systems as well as for systems that make use of ASR output. The productive morphology of Turkish yields many unique word forms resulting in the vocabulary growth problem. Figure 2.5 illustrates this problem. A word corpus with 182.3 M word tokens (units) and 1.8 M word types (distinct units) is used to obtain the vocabulary growth curves. A morphological analyzer [18] is used in order to obtain the root forms. The unparsed words are counted as single roots. The details of the data and the morphological analyzer will be explained in Chapter 3. In Figure 2.5, the number of distinct units increase significantly with the increasing amount of data for words. However, the curve for roots is almost leveled off at 901.2 K distinct units and demonstrates the word generation process with suffixation. On average 204 words are generated from each root and each word is decomposed of on average 1.7 morphemes including the root. The maximum number of words generated from a root is counted as 3348 for the verb "`etmek`" (to do). The maximum number of morphemes attached to a root is counted as 8, i.e "`ruhsat-lan-dır-ıl-ama-ma-sı-nda-ki`". This word occurs only once in the text data.

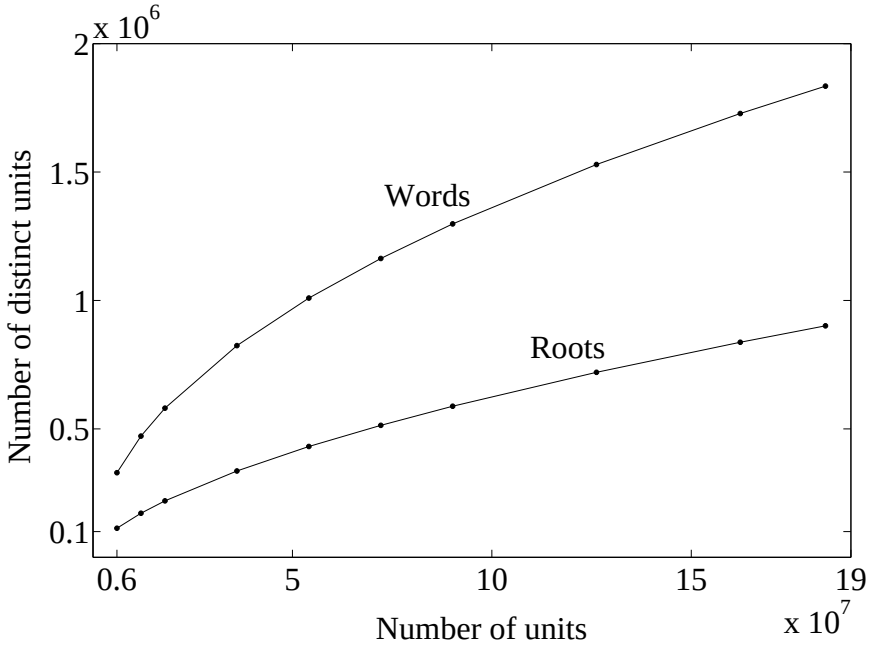


Figure 2.5. Vocabulary growth curves for words and roots.

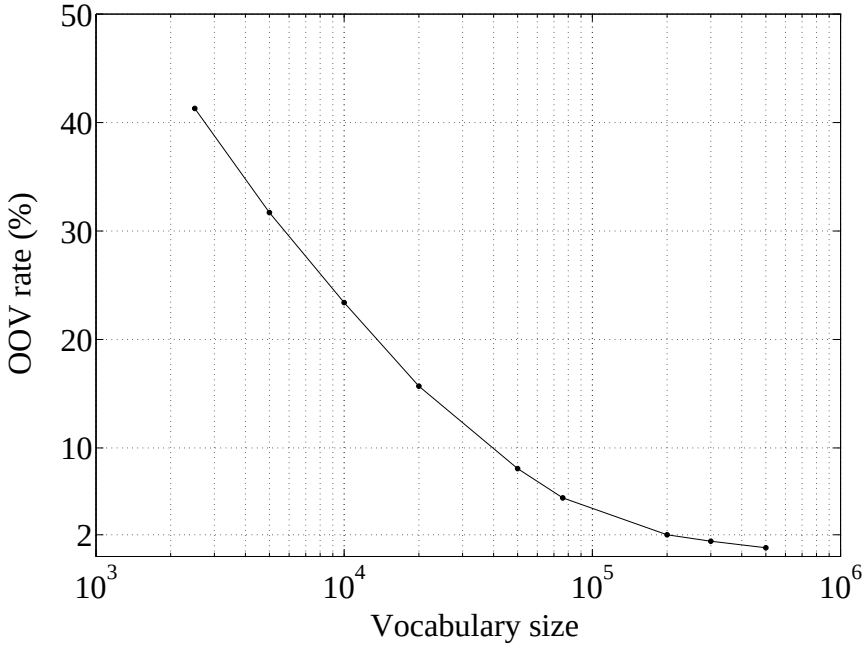


Figure 2.6. OOV rates for Turkish with different vocabulary sizes.

As was explained in Section 2.1, ASR systems utilize fixed and finite vocabularies. If a word is not in the recognition vocabulary (an OOV word), it has no chance to be recognized correctly and as a rule of thumb an OOV word brings up on average 1.5 recognition errors [19]. So, high OOV or low coverage rates directly translate into high WERs.

The main challenge of Turkish introduced to ASR is the high OOV rates since the vocabulary growth makes it difficult to have a vocabulary with high coverage. The OOV rates for different vocabulary sizes are given in Figure 2.6. The same word corpus with the vocabulary growth curves is used to generate the vocabulary. The most frequent N words are utilized as the vocabulary composed of N words. The OOV rates are calculated on a test data which contains 23.4 K words. It was shown that with an optimized 60K lexicon for English the OOV rate is less than 1.0 per cent for North American Business news [20]. However, even for vocabulary sizes that would be considered as large for English, the OOV rates for Turkish are quite high. Other morphologically rich languages[4] such as Finnish, Estonian, Hungarian and Czech also suffer from high OOV rates. Here are the published OOV rates for these languages;

[4] What we mean by morphologically rich languages is the languages having a complex morphological structure including agglutinative and highly inflectional languages.

15 per cent OOV with a 69 K lexicon for Finnish [21], 10 per cent OOV with a 60 K lexicon for Estonian [22], 15 per cent OOV with a 20 K lexicon for Hungarian [23] and 8.27 per cent OOV with a 60 K lexicon for Czech [24]. Here, the baseline systems are not directly comparable with each other, however, these numbers state that high OOV rates is a major problem for morphologically rich languages. Therefore, dealing with this OOV problem is crucial for the performance of the ASR systems.

A commonly proposed solution to high OOV rates in morphologically rich languages is to use sub-lexical recognition units instead of words. In this thesis, we also investigate sub-lexical approaches for Turkish. The advantage of sub-lexical units is that they can cover the language with a moderate vocabulary size. However, they introduce a new problem: generating non-word sequences in the recognition output. For instance, consider roots and suffixes as the vocabulary items. The recognition output will contain a sequence of roots and suffixes. Suffixes need to be attached to the roots in order to generate the words. The generated words should obey the vowel harmony and consonant harmony rules, however the language model itself does not guarantee the harmony of the sounds. For instance, the ungrammatical word "yıldızde" can be generated from the recognition output if the root "yıldız" and the suffix "-de" are recognized consecutively. The correct form of this word is "yıldızda" which obeys the vowel harmony rule. The last vowel of the stem and the first vowel of the morpheme in the word "yıldızde" are not compatible with each other. Therefore, vowel harmony and consonant harmony rules introduce a challenge to the ASR systems but only when sub-lexical recognition units are used.

The free word order is another challenge for ASR systems. The relatively free word order causes sparse data and sparse data leads to non-robust n-gram language model estimates.

2.3. Related Work

This thesis aims to investigate novel language modeling approaches for LVCSR of Turkish. Our proposed approaches are motivated by the challenges of Turkish in-

troduced to ASR systems. Before moving to the details of our research, the following sections will review the related work on sub-lexical language modeling units in ASR, handling the OOV problem in word-based ASR and advanced language modeling approaches. At the end of the section, we will mention the previous studies on Turkish language modeling for ASR.

2.3.1. Sub-lexical Language Modeling Units

This section reviews sub-lexical language modeling units explored in ASR systems. Morphologically rich languages suffer from large number of OOV words in LVCSR applications. Consequently, sub-lexical recognition units are proposed for these languages to handle the OOV problem caused by using words as recognition units. The proposed sub-lexical approaches are classified as being grammatical or statistical according to the underlying algorithm utilized in splitting words into sub-lexical units. Grammatical sub-lexical units are obtained with rule-based morphological analyzers and statistical sub-lexical units are obtained with statistical approaches. Note that language modeling with sub-lexical units requires higher order n-grams than words to achieve the same context [25].

Morphemes, stems and endings are the examples of the grammatical sub-lexical units. These units have been utilized in language modeling of agglutinative languages like Korean, Finnish, Estonian and Hungarian and highly inflectional languages like Czech, Slovenian and Arabic. Language modeling based on morphemes and roots was explored in [26] for a small vocabulary ($\approx$3 K) ASR task of German. Morpheme-based language model resulted in slightly better accuracies than word-based language model, however, utilizing only roots in language modeling degraded the recognition accuracy. Morphemes were also utilized in language modeling for Czech ASR [27]. However in this work, morphemes did not yield any improvement over words most probably due to utilizing the same order (bigram) language models both for words and morphemes in the first-pass recognition. Morpheme-based language models were also utilized for ASR of agglutinative Korean language [28]. To deal with the coarticulation problem rising from very short morphemes, frequent and short morpheme pairs were merged before language

modeling. Merged morphemes outperformed words when same size vocabularies (32 K) were utilized in trigram language models. Morphology-based language modeling was also applied to Arabic and yielded better results than words [29, 30]. An important result obtained in [30] was that the language model built with a large word vocabulary (≈64 K) outperformed the morpheme model absolutely by 2.4 per cent, whereas, the language model built with a huge word vocabulary (≈800 K) achieved almost the same result with morpheme language model. Morpheme-based language models were also investigated for ASR of agglutinative Estonian language [31]. Language model built with a 60 K morpheme vocabulary outperformed word language model built with the same size vocabulary. Imposing length constraints on morphemes to eliminate very short units further improved the accuracy.

Stem+ending based language modeling was proposed for inflected languages in [32]. Endings are formed with the concatenation of suffixes, as a result, they are longer units than morphemes. Stem and endings were utilized as language modeling units in Slovenian LVCSR [33]. Since stem+ending model was less constrained compared to the word model, it led to an increase in the search space of the decoder. To achieve a comparable search space with words, a new search algorithm was proposed to reduce the search space of stems and endings by including features of the inflectional language into the design of the algorithm, e.g., restricting the correct stem and ending order, limiting the number of endings for an individual stem.

Developing new statistical algorithms to obtain sub-lexical units have become very popular recently. One of the earliest works in this area was presented in [34]. In this research, morpheme boundaries in a word were explored by using the frequency of the letter following a previous set of letters with the assumption that the predictability of a letter will decrease at the morpheme boundaries. Data-driven algorithms based on probabilistic models as well as some heuristics have been developed for unsupervised morpheme discovery in the last 10 years. One of the algorithms with publicly available software is Linguistica [35] that utilizes Minimum Description Length (MDL) principle to model unsupervised learning of the morphological segmentations. This work aims to find the segmentations as close as possible to the true morphemes and it is restricted

to only European languages, e.g., English, Spanish, in which the average number of affixes per word is less than agglutinative languages. Another work was motivated by the productive morphology of Russian [36]. This algorithm aims to obtain sub-lexical units (called particles) that maximizes the likelihood of the training data using the bigram particle language models with a greedy search approach. In contrast to the algorithm given in [35], this algorithm does not aim to find true morphological segmentations. Instead, it searches for meaningful units in terms of language modeling. Morfessor algorithm [37, 38], with publicly available software, is proposed for unsupervised discovery of morphemes. This algorithm is inspired by an earlier work by [39] that explores the word discovery during language acquisition of young children. In that respect, the work in [39] proposed a probabilistic model based on MDL principle to recover word boundaries in a natural raw text from which they have been removed. Morfessor algorithm also utilizes the MDL principle while learning a representation of the language in the data as well as the most accurate segmentations. Additionally, Morfessor algorithm is better suited for highly inflectional and agglutinative languages than Linguistica since it is designed to deal with languages with concatenative morphology. In [40, 41], basic phonetic knowledge of the language was incorporated into the Morfessor algorithm to improve the segmentations.

The annual Morpho Challenge[5] competitions which motivate unsupervised learning of the morphology help the development of new algorithms for sub-lexical units since 2005. Morfessor Algorithm has been utilized as the baseline statistical sub-lexical approach in Morpho Challenge tracks. Paramor [42], which tries to construct affixes closely mimicking the paradigms of a language, has competed against other algorithms in Morpho Challenge tracks for the last few years.

Statistical sub-lexical units have been explored in language modeling of highly inflected and agglutinative languages. Morfessor was explored on Finnish [21, 22, 43], Estonian [22] and Hungarian [23]. Morpheme-based language models were compared with the language models built with Morfessor segmentations for Finnish [21] and Hungarian [23]. Finnish ASR experiments were performed on two different tasks. Sta-

[5] www.cis.hut.fi/morphochallenge{2005, 2007, 2008, 2009}

tistical and grammatical units resulted in almost the same accuracy in book reading task. However, statistical units outperformed grammatical morphemes in news reading task where the number of foreign words that could not be handled by the morphological analyzer were quite high. In Hungarian ASR experiments for spontaneous speech, the best result was obtained with statistical segmentations. See [44] for the advances in building efficient speech recognition systems with the Morfessor segmentations. Additionally, an optimized sub-lexical approach was proposed in [45] for Finnish dictation and German street names recognition tasks. The Morfessor segmentations obtained with basic phonetic knowledge were explored in Amharic, which is a highly inflectional language, ASR [40, 41]. Language model built with the sub-lexical units obtained with the modified Morfessor algorithm outperformed the word language model.

2.3.2. Handling the OOV Problem in Word-based ASR Systems

This section reviews the approaches proposed for handling the OOV problem in word-based ASR systems. Large number of OOV words is a major problem for morphologically rich languages. As a result, sub-lexical recognition units instead of words are proposed for these languages. However, existence of OOV words in ASR systems is also a problem for other languages that are considered to be trivial compared to morphologically rich languages, such as English. In those languages, due to the higher coverage rate of the recognition vocabulary, incorrectly recognized OOV words do not significantly degrade the overall recognition performance. However, in some applications such as spoken dialog systems developed for buying flight tickets and voice mail transcription systems, recognizing the proper names incorrectly has a higher cost than the introduced WER. Therefore, handling the OOV problem is crucial for ASR systems of all languages. The approaches reviewed in this section are based on a multi-pass speech recognition strategy that aims to dynamically control the word vocabulary between successive passes in order to recognize OOV words correctly. In the first set of approaches, the main motivation is to detect the candidate OOV words or some higher level trigger words that help adapting the vocabulary for the second-pass recognition. In the second set of approaches, the main motivation is to select a subset of words from a large fallback vocabulary for the adapted vocabulary using the first-pass recognition

output.

OOV detection approach tries to detect OOV words in the first-pass recognition using a filler model and then tries to recognize those words correctly in the second-pass recognition by altering the baseline vocabulary. This method was used to handle OOV words caused by rare city names in Mercury flight reservation and Jupiter weather information systems [46]. SpeM [47], a tool that can extract words and word initial cohorts from phone graphs on the basis of a large fallback vocabulary, was used to modify the baseline vocabulary. Multi-pass OOV handling approach was also utilized to make use of the contextual information within the utterance itself to trigger the addition of new words to the vocabulary [48]. For instance, recognizing the state names correctly in the first-pass recognition triggers the addition of new city names associated with those states into the recognition vocabulary. In addition, word/phone hybrid language models were proposed for better OOV detection [49].

Instead of detecting candidate OOV words or higher-level triggers in the first-pass recognition, the recognition vocabulary can be directly adapted using the first-pass recognition output as the prior information. In this approach, the baseline vocabulary is modified with the fallback vocabulary words similar to the hypothesized words. This technique was applied to Czech speech recognition using morphology-based similarity [50] that considers two words similar if they have the same stem but different endings. Dynamic adaptation of the vocabulary is a major issue in Broadcast News transcription since the topic of each show can change regularly [51]. It has been shown that adapting the baseline vocabulary with the verbal inflections in addition to written news words did better in coverage of the test data than adapting the vocabulary only using the written news [52]. In addition to morphology-based similarity, the distance in terms of word co-occurence patterns between hypothesized words and a vocabulary database was used to select new vocabulary words which are relevant to the content of the input speech [53]. Moreover, in [54] it was shown that names are the biggest portion of the OOV words in Broadcast news and candidate transcriptions were selected from a list of names with phonetic-distance-based pruning using the ASR hypotheses.

In addition to adapting the recognition vocabulary using similar words to the hypothesized words, the search space of the decoder can be restricted with Hypothesis Driven Lexical Adaptation (HDLA) [55]. This is a special case of vocabulary adaptation. In HDLA, lattice output of the first-pass system was extended with similar words from the fallback vocabulary for each utterance. Second-pass recognition was performed with the extended lattice and an utterance specific language model built with adapted vocabularies. Several similarity measures have been defined to find the similar word pairs for extension. Significant improvements were achieved using the morphology-based [55] and the phonetic-distance-based [56] similarity criteria in adapted vocabularies for a Serbo-Croatian ASR task. In addition, grapheme distance based similarity criterion and artificial generation of the fallback vocabulary with language specific morphological inflection ending rules [57] were proposed.

2.3.3. Advanced Language Modeling

n-gram language models utilize word probabilities conditioned on n-1 previous words in calculating the probability of the word strings. n-grams are the most common language modeling approach in the state-of-the-art ASR technology due to their simplicity with their substantial recognition performance. However, they have some known deficiencies, e.g., non-robust estimates for sparse data, ignoring the deep structure of language, estimating language model parameters with generative methods. Therefore advanced language modeling approaches have been proposed to deal with the deficiencies of n-gram models.

In this section, language models are categorized in three different dimensions; (1) context, (2) model structure and (3) estimation method. The first dimension determines the context in predicting the language models. This context can be the history of the n-grams or the whole sentence. History-based language models utilize the conditional word probabilities, conditioned on the histories, in predicting the next word and sentence probabilities are estimated with the product of these conditional probabilities. On the other hand, whole sentence language models directly predict the sentence probability or score. The second dimension, model structure, determines how the relevant

information is incorporated into language modeling. Language model structure can be conventional or feature-based. Conventional language models incorporate the relevant information into the history of the n-grams. Feature-based language models encode the relevant information as a set of features. The advantage of feature-based models to conventional models is that they allow for easy integration of arbitrary knowledge sources into language modeling. The third dimension determines the method in estimating language model parameters which are the conditional word probabilities in conventional models and the feature weights in feature-based models. Generative and discriminative methods have been utilized in language model parameter estimation. Discriminative methods yield improved parameter estimates compared to generative methods by taking negative examples into account as well as positive examples in parameter estimation. We will review some major language modeling approaches related to our research in the following paragraphs.

The traditional word n-grams are history-based language models with a conventional language model structure and they are estimated with a generative method, MLE. For instance in bigram language models, the probability of a word string is estimated as follows:

$$P(W) \;=\; P(w_1\ w_2\ \ldots\ w_N) = \prod_{k=1}^{N} P(w_k|w_1 \ldots w_{k-1}) \approx \prod_{k=1}^{N} P(w_k|w_{k-1}) \quad (2.9)$$

Class-based language models [58] are the examples of history-based, conventional and generative models. The aim in class-based language modeling is to handle data sparseness by grouping words that have similar syntactic or semantic properties. In bigram class-based language models, the probability of a word string is estimated as follows where c_k represents the class of the k'th word w_k.

$$P(W) \approx \prod_{k=1}^{N} P(c_k|c_{k-1})P(w_k|c_k) \quad (2.10)$$

Structured language models [59] and probabilistic top-down parsing in language modeling [60] are examples of history-based, conventional and generative language

models. These models incorporate syntactic information into language modeling to capture the long distance dependencies. For instance, consider the sentence "`the dog heard yesterday barked again`"[6]. Here, the probability of the word "`barked`" will be estimated from the previous words "`heard`" and "`yesterday`" in 3-gram language models. However, "`barked`" needs to be predicted from the word "`dog`" in history-based models and this is the main motivation in the proposed approaches. In these models, the conditional word probabilities, $P(w_k|w_1 \dots w_{k-1})$, are calculated using the hierarchical syntactic information derived with left-to-right parsing.

Super ARV language model [61] is another example of history-based, conventional and generative language models. This model utilizes Constraint Dependency Grammar (CDG) to incorporate syntactic information into language modeling. In CDG, structural analysis is represented as assignments of dependency relations to functional variables associated with each word. Therefore, superARV model tightly integrates structural constraints at the word level. SuperARV language model estimates the joint probability of words $(w_1 \dots w_N)$ and their superARV tags $(t_1 \dots t_N)$ obtained from CDG as given in Equation 2.11. The conditional probability distributions are estimated with recursive linear interpolation among probability estimations of different orders.

$$P(w_1 \dots w_N t_1 \dots t_N) \approx \prod_{k=1}^{N} P(t_k|w_{k-2}w_{k-1}t_{k-2}t_{k-1})P(w_k|w_{k-2}w_{k-1}t_{k-2}t_{k-1}t_k) \quad (2.11)$$

Another linguistic knowledge integration technique in language modeling is the Factored Language Model (FLM) [62]. FLM is an example of history-based, conventional and generative language models. The basic idea in FLMs is to decompose words into a set of factors, i.e., $w_i = f_i^1 \dots f_i^K$, and to use these factors in backoff language models. In bigram FLMs, the probability of a word string is estimated as follows:

$$P(W) \approx \prod_{k=1}^{N} P(f_k^1 \dots f_k^K | f_{k-1}^1 \dots f_{k-1}^K) \quad (2.12)$$

[6] This example was taken from [59].

Factors such as morphological and syntactic information have been utilized as the conditioning variables for predicting the next word. Generalized backoff strategy enables the robust estimation of language model parameters with many conditioning variables. FLMs result in better probability estimates than traditional word n-grams especially when the training data is sparse. FLMs are similar to class-based models, however, there can be simultaneous class assignments, i.e., the factors, in FLMs in contrast to class-based language models.

Conditional exponential models have been proposed to combine the history-based modeling with the feature-based approach. Maximum entropy (ME) language model [63] is the most popular example of this type. In the ME model, the conditional probabilities are calculated with an exponential model, given as follows:

$$P(W) \approx \prod_{k=1}^{N} P(w_k|h_k) \tag{2.13}$$

$$P(w|h) = \frac{1}{Z(h)} \exp^{\sum_i \lambda_i f_i(h,w)} \tag{2.14}$$

Here, $Z(h)$ is the normalization term and $f_i(h, w)$ is an arbitrary feature as a function of word w and the history, h. If w represents the k'th word w_k, then history represents the word sequence $w_1 \dots w_{k-1}$. For instance, one of the features in the ME model can be defined as follows:

$$f_i(h, w) = \begin{cases} 1 & \text{if } w \text{ is "cat" and } h \text{ contains "the"}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.15}$$

λ_i's are the model parameters associated with the features and they are estimated with generative methods during language model training. In language modeling with ME model, trigger features were introduced to adapt the model expectations to the topic of discourse [63], semantic dependencies and syntactic structure were combined for language modeling for ASR [64] and semantic analysis was integrated into language modelling for spoken dialogue systems [65]. A joint morphological-lexical language model which is based also on ME modeling was proposed in [66] for morphologically

complex Arabic language.

Whole sentence exponential language models were proposed in [67] to model the global sentence-level phenomena rather than to model only the conditional structure. With this model, lexical and syntactic information at the sentence level is incorporated into language modeling. In this model, the probability of a word sequence is represented as follows:

$$P(W) \approx \frac{1}{Z} P_0(W) \exp^{\sum_i \lambda_i f_i(W)} \tag{2.16}$$

Here, Z is a global normalization constant that depends only on λ_i's, $P_0(W)$ is the initial probability of the sentence and $f_i(W)$ can be an arbitrary sentence-level or history-based feature. Note that Z eliminates the calculation of the normalization term for each history in Equation 2.14. For instance, one of the sentence-level features can be defined as follows:

$$f_i(W) = \begin{cases} 1 & \text{if the subject of the sentence is "I",} \\ 0 & \text{otherwise.} \end{cases} \tag{2.17}$$

Global linear models, e.g., perceptron algorithm, global conditional log-linear model, have been successfully applied to language modeling tasks for speech recognition [68, 69, 70]. Language modeling with these models are the examples of whole sentence, feature-based models with discriminatively trained feature parameters and they are called discriminative language models. The perceptron model defines a cost on the word sequence also as a function of the acoustic observations, given as follows:

$$F(W, A) = \lambda_0 \log P(W, A) + \sum_i \lambda_i f_i(W) \tag{2.18}$$

Here, the features can be defined over the whole sentence as given in Equation 2.17. $\log P(W, A)$ represents the logarithm of acoustic and language model scores assigned to the word sequence W. Global conditional log-linear model defines a conditional

probability distribution over W, given as follows:

$$P(W|A) = \frac{1}{Z(A)} \exp^{F(W,A)} \tag{2.19}$$

$Z(A)$ is a normalization constant that depends on A and λ's, and makes the sum of the probability distribution equal to one. In these two models, the model parameters, λ_i's, are learned discriminatively through the training examples. In discriminative language models, training examples contain the positive examples (reference transcriptions) as in the generative models and also the negative examples (ASR errors). Discriminative language models have been demonstrated to consistently outperform generative modeling approaches, partly due to improved parameter estimation and partly due to the ease with which many overlapping features can be included in the models. In [68, 69], utilizing only the word n-grams as features outperformed traditional generative n-gram approach and incorporating syntactic or morphological features to n-gram features yielded additional improvements [70, 71]. Trigger-based features were also incorporated into discriminative language modeling [72] in order to use the conversation context as an additional information source. In addition to language modeling, discriminative training have been successfully applied to acoustic modeling [73, 74]. Discriminatively trained acoustic and language models can be utilized in the same system, however common approaches optimize the parameters of these two components independently. Discriminative training of decoding graphs were proposed in [75] to perform a combined model optimization for ASR components and this work was extended to LVCSR in [76].

Using the advanced language models in the first-pass recognition is not as easy as the conventional n-grams even when linguistic information is tightly integrated into the n-gram structure. Therefore, the language modeling approaches reviewed in this section, except discriminative training of decoding graphs, have been utilized in ASR via rescoring or reranking frameworks. In rescoring, the baseline language model score is replaced with the improved language model score on the ASR output to obtain the most likely hypothesis. In reranking, the baseline model generates N candidate hypotheses for each utterance and these hypotheses are reranked with a set of features

to obtain the best scoring hypothesis.

2.3.4. Turkish Automatic Speech Recognition

The ASR research for Turkish has grown rapidly in the last 10 years. The grammatical sub-lexical units, their combinations and statistical sub-lexical units have been utilized in language modeling of Turkish for ASR.

Morpheme-based language models as well as lattice extension with HDLA were proposed in [77]. Due to the ambiguous parses in the available morphological analyzers, syllables were utilized instead of grammatical morphemes as language modeling units. The short language modeling units problem encountered with syllables were handled by merging syllables to obtain longer units with word-positioned syllable classes. This approach solved the OOV problem, however, it did not yield any improvement over the word language model built with a 30 K vocabulary. HDLA was only explored in terms of OOV handling performance and no ASR results were reported.

The groupings of grammatical morphemes, called inflectional groups, were proposed as language modeling units in [78]. This research was based on estimating the word probability from it's inflectional groups. Following the work given in [78], extension of inflectional groups to n-gram language modeling as well as utilizing stem+ending models for Turkish were proposed in [79]. However, these models were not evaluated in terms of recognition accuracy or OOV handling.

The work in [80] presented a comparative study of morpheme, stem+ending and syllable language models in terms of generalization capacity of language models and OOV handling. The ASR experiments were also reported for these sub-lexical units, however, for a small vocabulary isolated word recognition task. This work was extended to continuous speech recognition in [81, 82] with a new model utilizing words, stem+endings and morphemes together in the same model. This combined model slightly outperformed the word model in terms of recognition accuracy when 10 K units were used in combined and word bigram language models.

In [83], bigram stem+ending model was compared with bigram stem model in terms of recognition accuracy in a small vocabulary ASR task. Stem model outperformed the stem+ending model when the language models were trained on a very small text corpus (<1 M words). However, stem+ending model was shown to outperform stem model when the text corpus size increased to approximately 6 M words [84].

Statistical sub-lexical units obtained with the Morfessor algorithm, was first applied to Turkish in [85]. Statistical units were shown to outperform words (60 K vocabulary) and grammatical units. In this research, language models were built on a text corpus containing only 2 M words.

FLMs were applied to Turkish in [86] using a small text corpus for language modeling. However, only determining the structure of FLMs, e.g., the set of conditioning variables, backoff procedure, was investigated by a data-driven search. The best FLM structure resulted in a more robust language model than words, however, recognition performance of FLMs was not investigated.

The common weaknesses in most of these previous studies are the small amount of training data both for acoustic and language models, small vocabulary sizes both in word and sub-lexical models and lower order n-gram models especially for sub-lexical units. Therefore, these models do not yield comparative baselines for a fair comparison of the proposed sub-lexical units. The most comprehensive previous research on language modeling for Turkish LVCSR was reported in [87]. In this work, the acoustic and language models were trained on much larger amounts of data, 34 hours of speech corpus and 81 M words text corpus. This work investigated words, stem+endings and syllables as language modeling units and compared their performances on an LVCSR task. Stem+ending model outperformed word and syllable models in recognition accuracy. Additionally, this work dealt with the over-generation problem of sub-lexical units by a post-processing approach addressing the vowel harmony rule of Turkish and further improvements were achieved over the best scoring stem+ending model.

3. DATA, TOOLS AND SYSTEM DESCRIPTION FOR TURKISH ASR

This chapter explains the acoustic and text data utilized in building Turkish LVCSR systems, the linguistic tools utilized in generative and discriminative language modeling research and the details of the Turkish LVCSR systems developed in this thesis for investigating our proposed language modeling approaches.

Performing ASR research for Turkish is a challenging task both due to the characteristic of the language and due to the lack of the publicly available language resources such as data and linguistic tools. DARPA EARS (Effective, Affordable, Reusable Speech-to-Text) programs have stimulated the improvements in robust ASR technology with the goal of developing speech-to-text systems with richer and much more accurate automatic transcriptions. Existence of shared language resources is very crucial in the ASR research for system training and development. Therefore, within the context of EARS, Linguistic Data Consortium (LDC)[7] provides the required data to the researchers. These data contain conversational and broadcast speech with reference transcripts for acoustic modeling, lexicons and texts for language modeling, and other types of complex annotation in all of the target languages which are English, Chinese and Arabic [88]. However, there were no publicly available language resources for Turkish until recently. In 2006, a standard phonetically-balanced Turkish microphone speech corpus [89] became available through LDC. Most of the researchers collect their own acoustic and text data and develop their own linguistic tools for Turkish ASR and Natural Language Processing (NLP) research.

In this research, we also collected our own acoustic data in addition to the text and the acoustic data provided to us by other researchers. Due to the availability of these data at different time intervals of this Ph.D. research, the performance of some of our proposed techniques were evaluated on different baseline Turkish LVCSR systems. The details of the acoustic and text corpora, linguistic tools and the baseline LVCSR

[7] http://www.ldc.upenn.edu/

systems will be explained in the following sections.

3.1. Acoustic and Text Data

3.1.1. Acoustic Data

We used two different acoustic corpora in this research. The first one contains the microphone recordings of read speech and it is called the Read Speech (RS) database. The second one contains the Broadcast News recordings and it is called the Broadcast News (BN) database.

The RS database is composed of the microphone recordings of the phonetically balanced utterances read by male and female native Turkish speakers (over 250 speakers). These data were collected in METU[8] and Sabancı University[9] and contain 17 hours of speech. See [89] and [87] for the properties of the data coming from METU and Sabancı University respectively. RS database was used to build the acoustic models for the newspaper content transcription system. Details of this system are given in Section 3.3.1. A separate test set was recorded for the evaluations. The test material consists of one hour (6989 words) of newspaper sentences read by one female speaker.

The BN database is composed of the recordings of the Broadcast News programs and their corresponding reference transcriptions. The BN database collection process started at Boğaziçi University in 2005 with BAP (project no: 05HA202) and TÜBİTAK (project no: 105E102) research projects. In this database, Broadcast News programs were recorded daily from a radio channel (VOA) and four different TV channels (CNN Türk, NTV, TRT1 and TRT2). Then these recordings were segmented, transcribed, and verified. Acoustic segmentations were generated from long speech signals by automatically segmenting the speech into smaller pieces according to acoustic information, pauses. The incorrect segmentations, wrong detection of pauses, were manually corrected. The transcription guidelines were adapted from Hub4 BN transcription

[8] Thanks to Tolga Çiloğlu for sharing the acoustic data with us.
[9] Thanks to Hakan Erdoğan for sharing the acoustic data with us.

Table 3.1. Amount of data for various acoustic conditions (in hours)

Partition	f0	f1	f2	f3	f4	fx	Total
Train	67.2	15.7	8.3	19.8	73.6	3.3	188
Held-out	1.1	0.1	0.1	0.5	1.3	0.0	3.1
Test	0.9	0.1	0.1	0.7	1.4	0.1	3.3

guidelines. The annotation includes topic, speaker and background information for each acoustic segmentation. Once the data were processed, the acoustic data were converted to 16kHz 16-bit PCM WAV format, and segmentation, speaker and text information was converted to the NIST STM format.

In this thesis, we used approximately 194 hours of speech from the BN database as the acoustic data. These data were partitioned into disjoint training (188 hours), held-out (3.1 hours) and test sets (3.3 hours). The reference transcriptions of the acoustic training data include 1.3 M words. This acoustic training data was utilized to build the acoustic models for the BN transcription system. Details of this system is given in Section 3.3.2. Table 3.1 gives the breakdown of the data in terms of acoustic conditions. Here classical Hub4 classes are used: (f0) clean speech, (f1) spontaneous speech, (f2) telephone speech, (f3) background music, (f4) degraded acoustic conditions, and (fx) other.

The BN database contains almost 11 times more speech data than the RS database. The test set of the BN database is more challenging due to the speaker variations in Broadcast News programs and the various acoustic conditions.

3.1.2. Text Data

In this research, we used two different text corpora for building statistical language models. The first one contains 26.6 M words collected from the web. 43.6 per cent of these text data come from various domains (See [82] for the details) and the

rest contains only the sports news[10]. These text data are called Text-I corpus. 26.6 M word tokens in this corpus result in 675 K word types from various topics. These text data were used to build the statistical language models for the newspaper content transcription system.

The second one, also collected from the web, contains 182.3 M words[11]. These data come from three major Turkish news portals. See [18] for the details of this text corpus. These text data are called Text-II corpus. 182.3 M word tokens in this corpus result in 1.8 M word types from various topics. These text data were used to build the statistical language models for the BN transcription system. Text-II corpus contains almost seven times more text than Text-I corpus. Note that Text-II corpus contains a more general content since sports news is very dominant in Text-I corpus.

3.2. Linguistic Tools for Turkish

3.2.1. Morphological Parser

Morphology determines the structure of word formation and morphological parsers use the word formation rules to decompose words into their component morphemes [90]. In this research, we utilized two different morphological parsers for Turkish. The first one was developed by Haşim Sak and the second one was developed by Kemal Oflazer. They are referred to as Sak's parser and Oflazer's parser respectively in this thesis. In Sak's parser, the morphophonemic rules and lexicon were adapted from the PC-Kimmo implementation of Kemal Oflazer, but implemented using Finite State Transducers (FSTs). There are also prolog-based morphological parsers developed for Turkish [91, 80].

Estimating a language model based on morphological units requires a morphological parser. We used Sak's parser in Chapter 4 to obtain the morphological language modeling units. See [18] for the details of this parser. An example output from Sak's

[10] This portion of the text data was collected in METU. Thanks to Tolga Çiloğlu for sharing the text data with us.

[11] Thanks to Haşim Sak for sharing the text data with us.

alın[Noun]+[A3sg]+[Pnon]+[Nom] *(forehead)*

al[Noun]+[A3sg]+Hn[P2sg]+[Nom] *(your red)*

al[Noun]+[A3sg]+[Pnon]+NHn[Gen] *(of red)*

al[Verb]-Hn[Verb+Pass]+[Pos]+[Imp]+[A2sg] *((you) be taken)*

Figure 3.1. Output of Sak's morphological parser with English glosses. Only 4 out of 8 possible interpretations are given.

alın+Noun+A3sg+Pnon+Nom *(forehead)*

al+Adj ^DB+Noun+Zero+A3sg+P2sg+Nom *(your red)*

al+Adj ^DB+Noun+Zero+A3sg+Pnon+Gen *(of red)*

al+Verb ^DB+Verb+Pass+Pos+Imp+A2sg *((you) be taken)*

Figure 3.2. Output of Oflazer's morphological parser with English glosses. Only 4 out of 6 possible interpretations are given.

parser for the word "alın" is given in Figure 3.1. The English glosses are given in parenthesis for convenience. The parser output shows the root, its part-of-speech (PoS) tag in brackets and lexical morphemes with associated morphological feature tags in brackets. The morphological feature tag without any morpheme indicates that this feature is applicable to the current word form. The capital letters in the lexical morphemes are used in two-level morphology to handle some phonetic modifications in the suffixation process such as vowel harmony and consonant harmony. The distinction between inflectional and derivational morphemes are denoted with "+" and "-" signs respectively. The derivational suffixes can change the PoS category of the derived word form, therefore the PoS tags are given in the morphological feature tags of derivational suffixes.

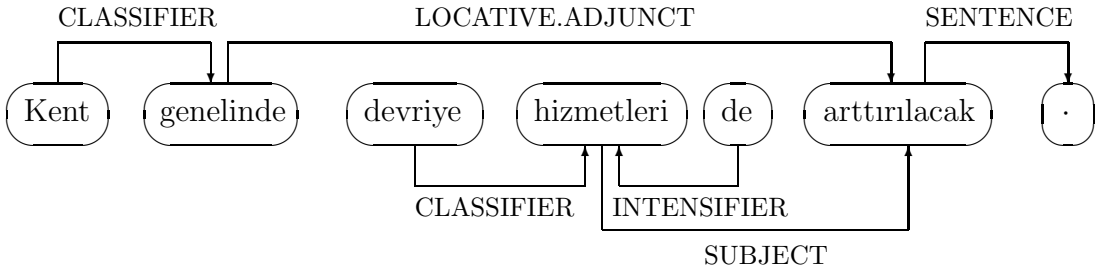
Oflazer's parser is the most famous and known Turkish morphological parser. In our research, it is used in Chapter 6 to obtain the morphological and syntactic features for discriminative language models. See [92] for the details of this parser. Fig. 3.2 shows the morphological segmentations of the same word, "alın", with Oflazer's parser. In

these segmentations, the root followed by its PoS tag and the other morphological feature tags are given. The lexical morphemes are not given in the segmentations. The ^DB labels denote the derivation boundaries and the following morphological tag shows the PoS of the derived word form.

The difference between the outputs of Sak's and Oflazer's parsers can be clearly seen from Figures 3.1 and 3.2. First, each parser yields different number of segmentations for the same word. Second, the lexical morphemes are missing in Oflazer's segmentation outputs. Third, the derivation boundaries are labelled differently in each parser. The feature tags in these two parsers are almost the same, except the tag `Zero`. However, due to the differences in the lexicon and the morphotactics, these parsers can yield different outputs for the same word. There is no direct comparison between the performances of these parsers. In this thesis, providing lexical morphemes makes Sak's parser more suitable for the language modeling research with morphological recognition units. However this parser's outputs are not compatible with the input expected by the available disambiguation tools and the dependency parser since these tools are expecting Oflazer's output. Therefore, we utilized Oflazer's parser in discriminative language modeling research where disambiguation tool and the dependency parser are required in extracting morphological and syntactic information from hypothesis sentences.

3.2.2. Morphological Disambiguator

The morphological parsing of a word, as shown in Figures 3.1 and 3.2, may result in multiple interpretations of that word due to complex morphology. This ambiguity can be resolved using morphological disambiguation tools for Turkish [93, 94, 95]. In this research, we used the perceptron-based morphological disambiguation tool to resolve the ambiguity in multiple parses of the words in the hypothesis sentences. See [95] for the details of this tool.



WORD:	Kent	genelinde	devriye	hizmetleri	de	arttırılacak	·
GLOSS:	The city	throughout	patrol	services	also	will be increased	·

Figure 3.3. Example dependency tree with Eryiğit's parser.

3.2.3. Dependency Parser

Syntax, the rules of sentence formation, determines the organization of the sentences with the constituent structure and the syntactic dependencies. Constituent structure refers to the hierarchical organization of the constituents of a sentence and syntactic dependencies specify the lexical dependencies in the sentence, for instance the presence of a word or a morpheme can depend on the presence of another word or a morpheme in the same sentence [90]. Syntactic dependencies are represented in terms of dependency trees that show the dependency links between head and dependent words. Dependency parsers try to find the underlying dependency tree in a sentence.

In this research, we utilized the dependency parser developed by Gülşen Eryiğit. This parser is called Eryiğit et al.'s dependency parser. This is a classifier-based deterministic parser utilizing Inflectional Groups (IGs) as the parsing units to find the dependency links between words. In Figure 3.2, the groupings of the consecutive morphological feature tags separated by derivation boundaries are called the IGs. See [96] for the details of this parser. The example Turkish sentence, which means *"Patrol services will also be increased throughout the city"*, is analysed with Eryiğit et al.'s dependency parser in Figure 3.3. The incoming and outgoing arrows in the figure show the dependency relations between the head and the dependent words with the type of the dependency. The words with English glosses are also given for convenience. In our research, this dependency parser was utilized to obtain the syntactic features for discriminative language models.

3.3. System Descriptions

In this thesis we developed two different LVCSR systems with the available data at different time intervals of this Ph.D. research. The first one is for newspaper content transcription and the second one is for Broadcast News transcription.

3.3.1. Newspaper Content Transcription System

The baseline acoustic model for this system was built using the RS database. Since Turkish is almost a phonetic language, graphemes were used in acoustic modeling instead of phonemes. The HTK [97] front-end was used to obtain the MFCC-based acoustic features. We used decision-tree state clustered cross-word triphone HMMs with approximately 5000 states as the acoustic model. Each HMM state had a GMM with six mixture components. AT&T tools [98] were utilized in acoustic model training. The baseline acoustic model was speaker independent.

The baseline language model for this system was built using the Text-I corpus. We used a moderate vocabulary size, the most frequent 50 K words in the text data, in language modeling. There is always a trade-off between the OOV rate and the system complexity. Moderate vocabularies result in higher OOV rates than large vocabularies, however, large vocabularies require more memory and computational power and a huge amount of text for robust language model estimates. Considering the amount of the text data (26.6 M words) and computer facilities of our laboratory at that time, 50 K was a reasonable vocabulary size for the newspaper content transcription system. As explained in Section 3.1.2, sports content is more dominant than the generic content in the text data. However, the test recordings do not contain any sports content. In order to eliminate the dominance of sports news in language modeling, two different n-gram language models were built: one with the generic content of the corpus and one with the sports content of the corpus. Then, these two models were linearly interpolated to reduce the effect of out-of domain data (the sports news). The interpolation constant was optimized to minimize the test set perplexity. The n-gram language models were built using the SRILM toolkit [99] with interpolated Kneser-Ney smoothing. Entropy-

based pruning [100] was also applied to the language models due to the computational limitations.

The performance of this system was evaluated on recordings of the newspaper sentences read by one female speaker. The 50 K vocabulary resulted in 11.8 per cent OOV rate on the test data. The OOV rate was calculated over the word tokens in the reference transcriptions. Speech recognition was performed with the AT&T decoder [101] using the baseline acoustic and language models. The recognition performance was evaluated in terms of WER. The newspaper content transcription system resulted in 38.8 per cent WER on the test data.

3.3.2. Broadcast News Transcription System

The baseline acoustic model for this system was built using the BN database. The same procedure with the previous system was followed for acoustic model training. In this system, the decision-tree state clustered cross-word triphone models had 10843 HMM states and each HMM state had a GMM with 11 mixture components.

The baseline n-gram language model with interpolated Kneser-Ney smoothing as well as entropy-based pruning [100] was built using the SRILM toolkit [99]. This language model was built using the Text-II corpus. We also utilized the reference transcriptions of the acoustic model training data in language modeling. The language models built with the Text-II corpus and the in-domain data (BN transcriptions) were linearly interpolated. The interpolation constant was optimized to minimize the held-out set perplexity. We could accommodate larger vocabularies in BN transcription system due to the improved computer facilities of our laboratory and increased amount of the text data (182.3 M words). Therefore, the most frequent 200 K words in the Text-II corpus and the in-domain data were selected as vocabulary items to balance the trade-off between the OOV rate and the system complexity.

The speech recognition performance of the BN transcription system was evaluated on the BN test data. The 200 K vocabulary resulted in 2.0 per cent OOV rate on the

test data. Speech recognition was performed with the AT&T decoder [101] using the baseline acoustic and language models. In order to reduce the effect of language model pruning on the recognition accuracy, the first-pass lattice outputs were rescored with an unpruned language model.

In the BN transcription system, we obtain two different baselines: one with acoustic segmentations and the other one with linguistic segmentations of the test data. The original BN data contain the acoustic segmentations where the speech signal is segmented into utterances using pauses. An acoustic segment in the BN data can contain speech from more than one sentence, for instance the speech belonging to the last two words of a sentence and the first five words of the next sentence. Linguistic segmentations are complete sentences and the speech signal is segmented according to linguistic information, sentence boundaries. We will need the ASR transcriptions corresponding to the linguistic segmentations in Chapter 6, since ASR output will be served as the input for the linguistic processing.

Sentence boundary detection is a preliminary step in many speech technologies requiring linguistic segmentations. Automatic sentence segmentation has also been proposed for Turkish [102]. In our research, we assume that the sentence boundary locations are given beforehand. We utilize the punctuations marking the sentence boundaries in the reference transcriptions to obtain the automatic transcriptions of linguistic segmentations from acoustic segmentations. The flow chart given in Figure 3.4 summarizes the main steps of our supervised approach.

In our approach, we first detect the acoustic segmentations containing sentence boundaries using their reference transcriptions. For the segmentations containing sentence boundaries, we find the sentence boundary locations using forced-alignment. In forced-alignment, the ASR system is forced to recognize the reference transcription, consequently, the time intervals corresponding to each letter in the reference transcriptions are provided in a neat way. Then the acoustic segmentations containing sentence boundaries are split into smaller pieces from the sentence boundaries. This approach guarantees that the new segmentations do not contain speech from different sentences.

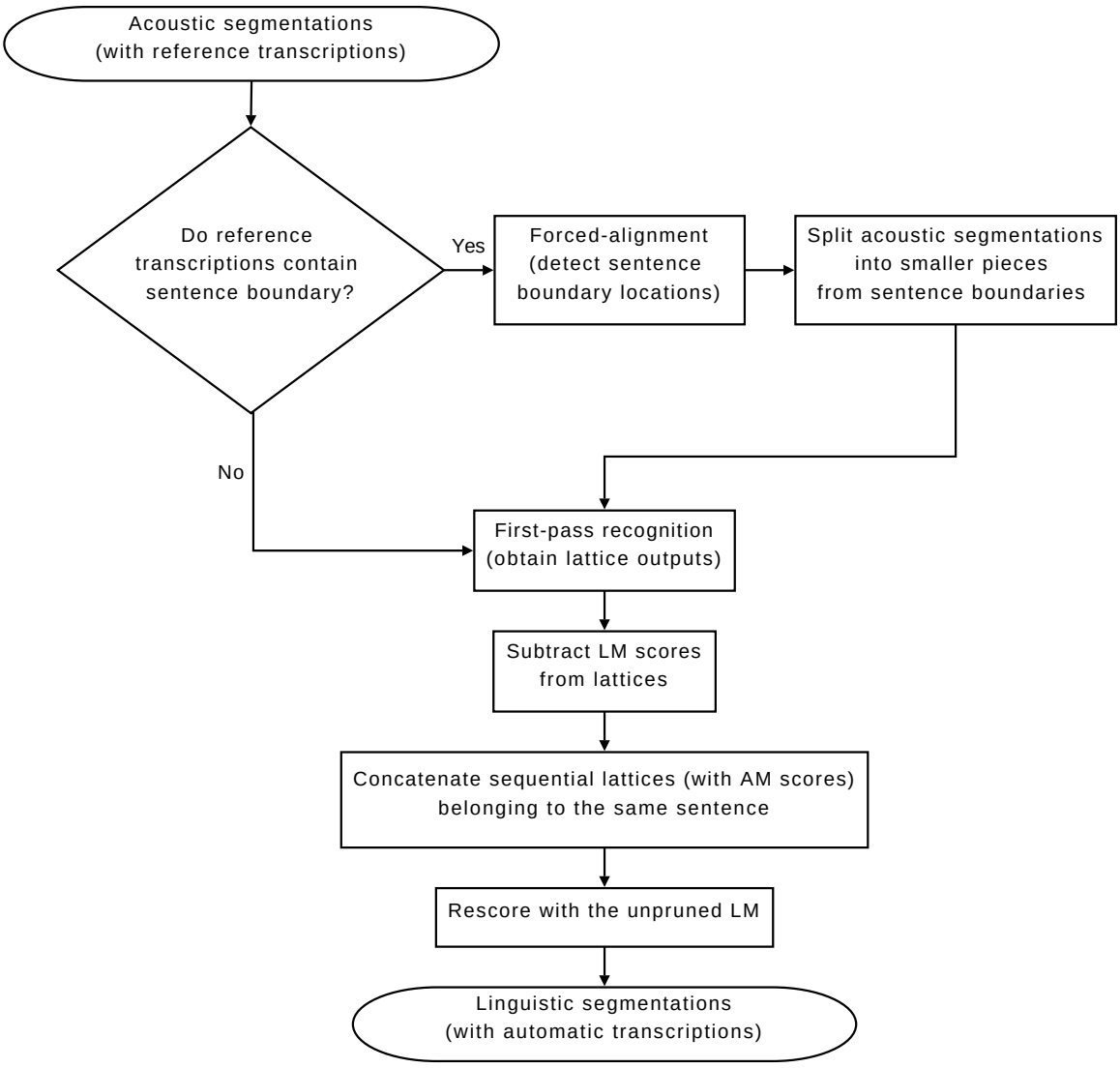


Figure 3.4. Flow chart showing the main steps in obtaining automatic transcriptions of linguistic segmentations from acoustic segmentations.

First-pass recognition is performed on these segmentations using the baseline ASR system. The lattice outputs are generated instead of 1-best transcriptions for rescoring purposes. Before rescoring, the baseline language model scores are subtracted from the lattices and the lattices containing sequential pieces from the same sentence are concatenated. This results in lattices of sentences containing only the acoustic model scores. After rescoring with the unpruned language model, automatic transcriptions of the sentences are obtained. We do not prefer generating the linguistic segmentations before the first-pass recognition, since this may correspond to very long speech segments and the ASR system can fail in finding the complete transcriptions of these sentences. This approach is only utilized in Chapter 6 to obtain the ASR transcriptions.

Table 3.2. ASR results with in-domain, generic and interpolated (generic + in-domain) language models.

Language Model (LM) Trial	WER (per cent)
generic + in-domain LMs (Baseline LM)	23.4
generic LM	25.2
in-domain LM	31.2

The baseline with acoustic segmentations results in 24.1 per cent WER and the baseline with linguistic segmentations results in 23.4 per cent WER on the test data. In the baseline with linguistic segmentations, we also utilized hesitation words, i.e., aaa, eee, mmm, in language modeling. 0.5 per cent of the improvement in the WER is coming from linguistic segmentations and 0.2 per cent improvement is coming from taking hesitation words into account in language modeling.

We also performed a set of ASR experiments to show the effect of linear interpolation on the recognition performance and a set of cheating experiments to demonstrate our lower WER bounds with OOV word handling and improved statistical n-gram language models. The results are reported for the system with linguistic segmentations. In order to investigate the effect of linear interpolation, the performance of the generic and the in-domain language models were investigated separately on the BN system by keeping the other ASR components the same. 200 K vocabulary that yields 2 per cent OOV rate on the test data was utilized in all the experiments. Table 3.2 summarizes the results for interpolated, in-domain and generic language models separately. The result with linearly interpolated language models, the baseline language model, is given on the first row of the table. Note that generic language model was built with 182.3 M words text corpus and in-domain language model was built only with 1.3 M words text corpus. The effect of the linear interpolation on the ASR performance can be clearly seen in the table. Utilizing only the generic or the in-domain language models degrade the performance according to the baseline language model. Linear interpolation improves the WER by 1.8 per cent on the generic language model and by 7.8 per cent on the in-domain language model. The worst performance is obtained with the in-domain

Table 3.3. ASR results for the cheating experiments.

Language Model (LM) Trial	WER (per cent)
generic + in-domain LMs (Baseline LM)	23.4
generic + in-domain LMs (no OOV words in the vocabulary)	22.3
generic + in-domain LMs (test data is already seen in LM)	14.9

language model most probably due to the inefficient amount of text data for robust estimation of the language model parameters.

Additionally, we performed two cheating experiments to demonstrate our lower WER bounds for the baseline system with linguistic segmentations. The results are given in Table 3.3. The first experiment investigates the effect of handling the OOV words by including all the OOV words in the test data to the recognition vocabulary. The second experiment investigates the effect of improving statistical n-gram language models where the reference transcriptions of the test utterances have been already seen in the language models. These cheating experiments yield our lower WER bounds after OOV handling and improved n-gram estimations respectively.

4. SUB-LEXICAL UNITS FOR STATISTICAL LANGUAGE MODELING

This chapter investigates sub-lexical units for statistical language modeling in the context of Turkish LVCSR. Using sub-lexical units in ASR is a common approach employed for agglutinative languages to handle the OOV problem caused by using words as vocabulary items. In this approach, the recognition vocabulary is composed of sub-lexical units instead of words. The sub-lexical unit vocabulary should be capable of covering most of the words of a language to address the OOV problem and can therefore lead to an improvement in recognition accuracy. Therefore, logical choices of word segments can be used as the sub-lexical units. These sub-lexical units should be meaningful for prediction in language models and they should have limited confusion due to over-generation. For instance if the letters are used as sub-lexical units, only a vocabulary of 29 letters of Turkish will cover all the words in the language. However, letters are not meaningful sub-lexical unit choices since they require very long histories for accurate language model predictions and they allow more confusable paths in decoding.

In agglutinative languages, words are formed by concatenation of stems and affixes. Therefore, grammatical units such as stems, affixes or their groupings can be considered as natural choices of sub-lexical units in ASR systems. See Section 2.3.1 for the review of grammatical sub-lexical units in language modeling for ASR. Grammatical sub-lexical units are obtained by using language dependent rule-based morphological analyzers. The splitting of words into sub-lexical units is straightforward with morphological analyzers. However, morphological analyzers may suffer from the OOV problem due to many proper names and foreign words that usually occur in news texts, since a limited root vocabulary is compiled in the morphological analyzers together with the morphotactic and morphophonemic rules. For instance, a Turkish morphological analyzer [18] with 54.3K roots can analyze 96.7 per cent of the word tokens and 52.2 per cent of the word types in a text corpus of 212M words with 2.2M unique words. Even

though stems and affixes are natural sub-lexical choices, the need for expert knowledge of the language makes them inapplicable to languages lacking morphological tools.

In order to handle the drawbacks of grammatical sub-lexical units, their statistical counterparts have been proposed. See Section 2.3.1 for the review of statistical sub-lexical units in language modeling for ASR. Statistical sub-lexical units are morpheme-like units. They are obtained with data driven approaches, usually in an unsupervised manner, instead of morphological analyzers. The main advantage of this model compared to grammatical models is that it does not require an expert knowledge of the language. Therefore, it can easily be applied to any language. However, splitting words into sub-lexical units is not trivial in statistical segmentations. The statistical morpheme-like units are not supposed to match with the exact grammatical morphemes for language modeling, however, they should meet the meaningful unit criteria. Therefore, different algorithms are investigated to obtain reasonable morpheme-like units with statistical techniques (See Section 2.3.1). These algorithms only require a raw text corpus to learn the word segmentations. However, a basic phonetic knowledge of the language can be used to improve the segmentations [40, 41].

In this chapter, we investigate grammatical and statistical sub-lexical approaches for Turkish LVCSR. The main difference of this study from previous sub-lexical language modeling research on Turkish is that all proposed techniques were compared in the same LVCSR system trained on large acoustic and text databases. In addition, we performed a manual error analysis on the best scoring system in order to find the source of the recognition errors. The details of the proposed techniques will be explained in the following sections.

We also performed three joint studies on sub-lexical language modeling. These are; (i) lexical form grammatical sub-lexical units were proposed in addition to the surface form representations of grammatical units, (ii) Morfessor algorithm was applied to Turkish, (iii) Morfessor algorithm with phonetic features was applied to Turkish. The first one was a joint work with Haşim Sak (published in [2]), the second one was a joint work with Mikko Kurimo and the researchers in the Adaptive Informatics Research

Centre at Helsinki University of Technology (published in [22, 103, 104]), and the third one was a joint work with Thomas Pellegrini (published in [105]). In this chapter, we will roughly mention these three approaches in addition to our original work.

4.1. Language Modeling Units

In this thesis, words, stem+endings, and statistical morphs were utilized as the recognition units. Figure 4.1 illustrates segmentations of the same Turkish phrase using different sub-lexical units. The grammatical units were obtained with Sak's morphological parser. The parser output was simplified by removing PoS tags and morphological features. Here, the distinction between inflectional and derivational morphemes is not taken into account and all morphemes are preceded with "-". The examples showing the lexical and surface form representations of the morphemes are denoted by *Lex* and *Surf* abbreviations. The parser only provides the lexical form representations of the morphemes and they are converted to surface form representations using a morphophonemic transducer. Details of the recognition units will be explained in this section. Note that, sub-lexical sequences need to be converted to word sequences to evaluate the WER after decoding. In order to facilitate this conversion, special symbols can be inserted into word boundaries in language modeling or the sub-lexical units can be marked with special labels, i.e., "-", as denoted in the morpheme and stem+ending examples in Figure 4.1.

Words: `derneklerinin öncülüğünde`
Morphs: `dernek lerinin öncü lüğü nde`
Morphemes:
Lex: `dernek -lArH -NHn öncü -lHk -SH -NDA`
Surf: `dernek -leri -nin öncü -lüğ -ü -nde`
Stem+endings:
Lex: `dernek -lArH-NHn öncü -lHk-SH-NDA`
Surf: `dernek -lerinin öncü -lüğünde`

Figure 4.1. Turkish phrase segmented into statistical and grammatical sub-lexical units.

4.1.1. Word based Model

Using words as recognition units is a classical approach employed in most state-of-the-art ASR systems. The word model has the advantage of having longer recognition units which results in better acoustic discrimination among vocabulary items. However the vocabulary growth for words is almost unlimited for agglutinative languages and this leads to high OOV rates with moderate size vocabularies. It has been reported that a text corpus with 40 M word tokens results in less than 200 K word types for English and 1.8 M and 1.5 M word types for Finnish and Estonian respectively [104]. The number of word types is 735 K for the same size Turkish corpus. See Figures 2.5 and 2.6 for the vocabulary growth and OOV problems for Turkish.

4.1.2. Grammatical sub-lexical units: Stem+endings

In this work, we used Sak's morphological parser outlined in Section 3.2.1 to decompose words into grammatical morphemes. As a consequence of the complex Turkish morphology, a word can yield several morphological analyses as given in Figure 3.1. Removing the morphological ambiguity can be crucial in dependency parsing and word sense disambiguation. However resolving the ambiguity in ASR may not be as crucial as in NLP applications. As was mentioned in Section 3.2.1, we can not disambiguate the multiple analyses of a word since Sak's parser outputs used in our experiments are not compatible with the input expected by the available disambiguation tools. In [1], building language models with all the ambiguous parses, with the parses with the least number of morphemes and with random parses for each word token and type were investigated using Sak's parser outputs. It was found that there is no significant difference between the first two methods and these are better than random. Therefore, due to its simplicity, the parse with the least number of morphemes was selected in building language models in our research. Additionally, in [106] it was shown that utilizing the parse with the least number of morphemes performed slightly better than utilizing the disambiguated parse in Turkish ASR.

In statistical language modeling there is a trade off between using short and long

units in language modeling. Long units, i.e., words, result in OOV problem especially for agglutinative languages. Short units, i.e., letters, can handle the OOV problem with a moderate vocabulary size but they may not meet the meaningful unit criteria. Morphemes or their groupings are natural choices for grammatical sub-lexical units. However, there can be very short units, as short as a single letter, in grammatical morphemes. Therefore, if grammatical morphemes are used for language modeling, they will require longer n-grams in language modeling and they may introduce more over-generated recognition outputs. Stem+endings have been proposed as a compromise between words and morphemes. They provide better OOV rate than words, and they may handle the drawbacks of morphemes. Therefore, we only utilized stem+endings in our research as the grammatical sub-lexical units.

To obtain the stem+endings, we first extracted the stem from the morphological decomposition and the remaining part of the word was taken as the ending. So we did not need to use a morphophonemic transducer to obtain the surface form endings. Stems and surface form endings (See Figure 4.1) were used to generate the language models. Segmenting the text corpora, Text-II corpora (See Section 3.1.2 for the details), into stems and endings yielded 263.2 M units with 901.2 K distinct stems and 43.7 K distinct surface form endings. Each word in the text data is segmented into 1.5 units on average. As was given in Figure 4.1, we do not need to keep the morpheme boundary labels in surface form endings, however, the ending labels need to be kept in language modeling to locate the word boundaries easily after recognition.

We also investigated lexical form stem+endings in language modeling on Turkish Broadcast News Transcription task [2] as a joint work with Haşim Sak. The main motivation in this work is that same stems and suffixes in lexical form may have different phonetic realizations due to phonological and morphophonemic phenomena in suffixation. For instance, the words, `evler` *(houses)* and `kitaplar` *(books)*, are decomposed into their morphemes as `ev-ler` and `kitap-lar` respectively. Although, both of the words have the same morpheme (-lAr[12]) in the lexical form, the vowel of the plural morpheme is modified according to the last preceding vowel of the stem during the

[12] 'A' is the lexical symbol realized as /a/ or /e/ in surface form.

suffixation process. Therefore, sub-lexical units in surface form may reveal many units that correspond to the same stem or morpheme groups in lexical form. In statistical language modeling with lexical form stem+endings, we aimed to capture the suffixation process better with lexical form endings. However, lexical form stem+endings introduced the problem of finding the correct pronunciation of a lexical form ending in decoding. In our research, we added all possible pronunciations of a lexical ending as the pronunciation variants in the lexicon. As was given in Figure 4.1, the morpheme boundary labels are required in lexical form endings in order to perform the correct lexical to surface form mappings. Note that this mapping produces surface forms regardless of morphotactics. Therefore, lexical form stem+ending model only handles the over-generation problem due to incorrect morphophonemics. So, there can be still invalid unit sequences in the recognition output.

4.1.3. Statistically derived sub-lexical units: Morphs

We used morphs, which are morpheme-like units obtained with the Morfessor Algorithm [38], as the statistical sub-lexical units in this thesis. Morphs were first investigated in our research during my 2005 summer visit to Helsinki University of Technology[13] and the findings of this summer visit and our further joint works for Turkish as well as for Finnish and Estonian were published in [22, 103, 104]. Morfessor Algorithm is a data driven approach based on the MDL principle which learns a subword vocabulary in an unsupervised manner from a training vocabulary of words. The main idea in Morfessor is to find an optimal encoding of the data with a concise lexicon and a concise representation of the corpus. The main advantage of this model compared to grammatical models is that it does not require an expert knowledge of the language. Therefore, it can easily be applied to any language.

The Baseline-Morfessor algorithm [38] was used with default settings to automatically segment the word types in the text corpus, Text-II corpora (See Section 3.1.2 for the details). For robustness, only the words occurring at least three times were used

[13] Thanks to Mikko Kurimo, Mathias Creutz and Teemu Hirsimäki for their help in building morph-based language models.

in training and the remaining words were segmented into morphs with the Viterbi algorithm using the initial segmentations. This resulted in 50 K distinct morphs with 257 M morph tokens. Each word in the text data is segmented into 1.4 units on average.

As was given in Figure 4.1, there are no markers that can be utilized in detecting word boundaries in the morph segmentations. In the morph-based model, we explored three different ways of locating word boundaries in order to facilitate morph sequence to word sequence conversion after recognition.

1. Use a word boundary morph, #,
   e.g., "`dernek lerinin # öncü lüğü nde`"
2. Use no markers for word boundaries,
   e.g., "`dernek lerinin öncü lüğü nde`"
3. Mark non-initial morphs with "-",
   e.g., "`dernek -lerinin öncü -lüğü -nde`"

In the first scenario, the word boundaries were marked with a special morph, shown as "#", called word boundary morph. All the morph types (50 K morphs) and the word boundary morph were utilized as the vocabulary items in building the language model. In the second scenario, no word boundary symbol was utilized in language modeling in the first-pass recognition. However, the recognition output was rescored with a new language model containing word boundary information, as in the first approach, to locate the word boundaries. Before rescoring, optional word boundary symbols were inserted to every node in the recognition lattice. In the third scenario, the non-initial morphs were labeled with "-" sign as in the grammatical models. Since a statistical morph is learned independent of its position in a word, it can occur both as an initial or a non-initial morph. Therefore, marking the non-initial morphs increased the vocabulary size from 50 K to 76 K. Note that the language model in the first scenario and the rescoring language model in the second scenario suggest higher order n-grams compared to the other scenarios due to the inserted word boundary morphs.

We also investigated enhanced Morfessor algorithm with phonetic features for Turkish LVCSR as a joint work with Thomas Pellegrini. All the properties used in

the Morfessor algorithm are based on written language and do not incorporate any "oral" properties that could be useful for ASR. The main idea in enhanced Morfessor algorithm is to incorporate a simple phonetic knowledge of Turkish into Morfessor in order to improve the segmentations. Two main modifications were made to enhance Morfessor: a phone-based feature, called 'DF' for distinctive feature, and a constraint called 'Cc' for confusion constraint. DF was directly incorporated into Morfessor's probability estimates and Cc was indirectly incorporated into Morfessor as a yes/no decision in accepting candidate splits. Both of these modifications aimed to reduce the number of confusible morphs in the segmentations by taking phonetic and syllable confusability into account. The DF property is language specific since its features depend on the phones of the language. Tables 2.1 and 2.2 were utilized to define vowel and consonant distinctive features for Turkish. Enhanced Morfessor algorithm with phonetic features was developed by Thomas Pellegrini. The details of this algorithm can be found in [41] and our joint work on Turkish was published in [105].

4.2. ASR Results

ASR experiments for various units were performed on the BN transcription system (See Section 3.3.2 for the details). For the recognition task of each sub-lexical unit, we utilized the same acoustic model, the same decoder and a unit specific language model. Note that each word in the training data was segmented into 1.5 stem+endings and 1.4 morphs on average. Therefore, sub-lexical units suggest higher order n-gram language models to have similar n-gram coverage with words. Higher order n-gram language models based on sub-lexical units have been shown to improve especially the recognition of unseen words in the training text [25]. In order to make a fair comparison of the proposed sub-lexical units, the real-time factor (RTF) for decoding was set to the same number (RTF $\approx$ 1.5) for each experiment. The vocabulary sizes, coverage ratios for the given vocabularies, average unit length (AUL) in each vocabulary, units per word (UPW) and the WERs are given in Table 4.1. The best results of the word, stem+ending and morph experiments are given in bold. When calculating word coverage using sub-lexical units, a word is considered as an OOV word if it can not be generated by any combination of a stem and an ending or an initial unit followed by

non-initial units. The ASR experiments were performed on the acoustic segmentations of the held-out and the test data.

For the word based model, the effect of the OOV rate was investigated by changing the vocabulary size. We conducted experiments with large (50 K, 76 K) and very large (200 K, 300 K, 500 K) vocabulary sizes. The aim of using 50 K and 76 K vocabularies was to compare the performances of word and statistical sub-lexical models having the same vocabulary sizes. The best results were obtained with the 3-gram language models. Increasing the vocabulary size from 50 K to 500 K increased the coverage by 7.3 per cent and decreased the WER by 5.7 per cent on the test set. The coverage was calculated over the word tokens. Significant relative improvements at the level of $p < 0.001$ as measured by the NIST MAPSSWE significance test [107] were obtained with the increased vocabulary size until 200 K. The improvement from 200 K to 300 K vocabulary was statistically significant at $p < 0.01$. No significant improvement was obtained with 500 K vocabulary compared to 300 K vocabulary on the held-out data and only 0.2 per cent improvement was achieved on the test data. The word model with 500 K vocabulary is highlighted as the best model in Table 4.1. However, the word model with 200 K vocabulary will be used as the baseline-word model to balance the trade-off between recognition performance and language model complexity in the further experiments.

In the stem+ending model, the most frequent 76 K and 200 K units out of 945 K surface form stem and ending types were utilized as the vocabulary items. The reason for selecting these vocabulary sizes was to make the stem+ending system comparable with the morph and the baseline word systems in terms of vocabulary size. The selected vocabularies covered 99.6 per cent and 99.8 per cent of the word tokens on the test set and reduced the WER to 23.2 per cent and 23.1 per cent respectively. The best result was obtained with 4-gram language models. No significant gain was obtained by increasing the vocabulary size from 76 K to 200 K. The stem+ending model with 200 K vocabulary is highlighted in Table 4.1 as the best model due to performing slightly better than 76 K vocabulary on the test data. The improvements obtained on the test set with stem+ending models were significantly better ($p < 0.001$) than 50 K,

Table 4.1. Results for different language modeling units (Real-Time Factor ≈ 1.5)

Recognition Units	Lexicon		UPW	n-gram	Coverage (per cent)		WER (per cent)	
	Size	AUL			Held-out	Test	Held-out	Test
Words	50 K	9.4	1.0	3-gram	92.7	91.9	29.9	29.4
	76 K	9.7	1.0	3-gram	94.9	94.6	27.7	27.0
	200 K	10.4	1.0	3-gram	98.0	98.0	25.5	24.1
	300 K	10.6	1.0	3-gram	98.7	98.6	25.1	23.9
	500 K	10.9	1.0	3-gram	99.1	99.2	**25.1**	**23.7**
Stem+endings	76 K	8.0	1.5	4-gram	99.7	99.6	24.1	23.2
	200 K	8.6	1.5	4-gram	99.8	99.8	**24.1**	**23.1**
Morphs (w/ WB morph)	50 K	7.0	2.4	5-gram	100	100	25.3	24.6
(w/o WB morph)	50 K	7.0	1.4	4-gram	100	100	24.7	23.9
(non-initials marked with "-")	76 K	6.7	1.4	4-gram	100	100	**24.1**	**22.9**

76 K and 200 K word models.

In the morph model, we utilized three ASR experiments with the scenarios proposed for locating the word boundaries in the previous section. First scenario, using a word boundary morph, utilized 50 K morphs and additionally the word boundary morph as the vocabulary items. In the text corpora, the ratio of morph tokens to word tokens was calculated as 2.4 including the word boundary symbol. This suggests higher order n-gram language models and the best result was obtained with the 5-gram language model. In the second scenario, using no word boundary label, first-pass recognition was performed with a 4-gram morph language model built without the word boundary symbol. Then, optional word boundary symbols were added to every node in the lattice and this lattice was rescored with a 5-gram language model containing the word boundary symbols. Third scenario, marking non-initial morphs with "-", increased the vocabulary size from 50 K to 76 K and a 4-gram language model built with this vocabulary was utilized in the experiments. The morph experiments show that different approaches for locating the word boundaries can effect the recognition accuracy significantly. The best result was obtained with the third scenario where a marker is attached to the non-initial morphs, also highlighted in Table 4.1. This model yielded significant improvements ($p < 0.001$) over all the word based systems.

We did not report the results with the lexical stem+endings and with the morphs obtained with the enhanced Morfessor algorithm in Table 4.1. These experiments were performed on a different BN transcription set-up where acoustic and language models were trained with smaller amount of data than the current models. Therefore, the results obtained with these two approaches are not directly comparable with the results reported in this section. The lexical stem+ending model was shown to outperform the morph model, absolutely 0.8 per cent [2], and the morph model obtained with the enhanced Morfessor algorithm was shown to outperform the morph model obtained with the original algorithm, absolutely 1.1 per cent [105]. This improvement was obtained with the distinctive consonant features integrated into the enhanced Morfessor algorithm.

Table 4.2. ASR results for the baseline systems

	WER (per cent)			
	Acoustic segmentations		Linguistic segmentations	
	Word (200 K)	Morph (76 K)	Word (200 K)	Morph (76 K)
Held-out	25.5	24.1	24.1	22.9
Test	24.1	22.9	23.4	22.4

Among the word models, the model with 200 K vocabulary was selected as the word baseline by considering the trade-off between the recognition accuracy and the language model complexity. According to the WER results reported in Table 4.1, we selected the 76 K morph model, non-initials marked with "-", as the baseline sub-lexical model. For the baseline word and morph ASR systems, we also performed recognition experiments using the linguistic segmentations as explained in Figure 3.4. The automatic transcriptions with linguistic segmentations will be used in Chapter 6 to serve as inputs to the linguistic tools in order to obtain linguistic features for discriminative language models. The results for these experiments are summarized in Table 4.2. The results for the acoustic segmentations are also given for comparison. Linguistic segmentations yield much better baseline results both for the word and morph language models. The improvements obtained with the linguistic segmentations are much more pronounced on the held-out set. This can be related to the characteristics of the acoustics segmentations on the held-out set, i.e., longer acoustic segments. The average acoustic segmentation length is 4.6 seconds for the held-out data and 3.7 seconds for the test data.

In the ASR experiments with the linguistic segmentations, the morph-based system outperforms the word-based system by 1.0% on the test data. When we aligned the word hypothesis sentences with the morph hypothesis sentences, we calculated the WER as 14.7%. This investigation reveals that word errors are not in the same locations in the hypothesis sentences obtained with the word and morph language models.

Figure 4.2 compares the test set WERs of the baseline word and morph models

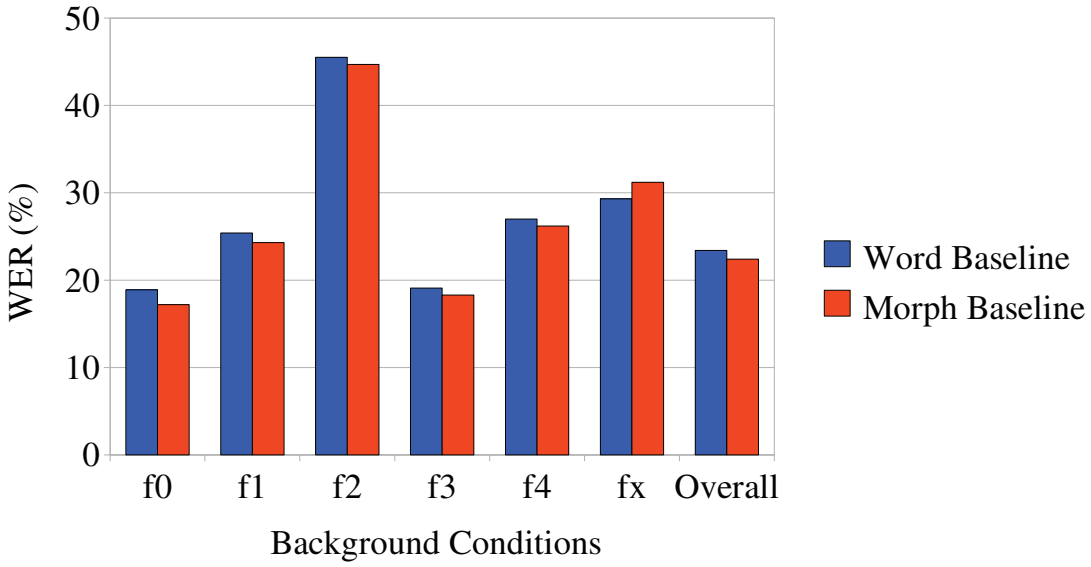


Figure 4.2. Comparison of WERs for the baseline word and morph systems according to the background conditions.

according to various acoustic conditions. The baselines with the linguistic segmentations were used in the comparisons. The Hub4 classes for acoustic conditions are as follows: (f0) clean speech, (f1) spontaneous speech, (f2) telephone speech, (f3) background music, (f4) degraded acoustic conditions, and (fx) all other speech. Sub-lexical baseline model outperformed the word baseline model in all the acoustic conditions except fx. Considering that fx has only six minutes of data in the test set, this is not a significant difference (See Table 3.1 for the break down of the test data in terms of acoustic conditions). Our baseline models have lower than 20 per cent WER on the clean speech and the improvement obtained by using morphs is 1.7 per cent absolute.

We conclude that sub-lexical units solve the OOV problem with smaller vocabulary sizes and outperform the word model in terms of the recognition accuracy. Note that the cheating experiment that investigates the effect of handling OOV words by incorporating all the OOV words in the test data to the recognition vocabulary yields 22.3 per cent WER (See Table 3.3). The baseline morph model yields 22.4 per cent WER on the test data with the same linguistic segmentations (See Table 4.2). This finding is a strong evidence of the superiority of morphs over words due to handling the OOV problem. In Table 4.1, the WER differences between the baseline sub-lexical

model and the word models are getting smaller with the increasing coverage in the word models. This shows that OOV problem can also be addressed by using very large vocabularies when huge language model training corpus is available. The same conclusion was achieved also for Arabic LVCSR [30]. In our experiments, the baseline morph model even outperforms the word model with 500 K vocabulary by 0.8 per cent (significant at $p < 0.001$). This suggests even larger vocabulary word-based ASR systems for Turkish to reach the same performance with the morph-based ASR system. However, it may not be possible to accommodate very large vocabularies and training data due to computational limitations.

4.3. Analysis of the Recognition Errors

Understanding the sources of recognition errors is very important to handle the shortcomings of the ASR systems. Therefore, this section investigates the recognition errors both at high and low levels. High-level error analysis only explores if the gains obtained with sub-lexical approaches are coming from dealing with OOV words. The low-level error analysis explores the type of the recognition errors, only on the morph-based model which is our best scoring model. In the low-level approach, the errors in the recognition output of the morph model were manually labeled according to some predefined error classes. This subjective labeling revealed important results that will be useful for our further research.

4.3.1. High-level Analysis: OOV or IV Word Errors

Sub-lexical recognition units have been proposed to address the OOV problem in morphologically rich languages. Therefore, we analyzed the improvements on OOV words to judge the effectiveness of each of the proposed sub-lexical approaches in the previous section. We defined new error metrics to evaluate the error rates for OOV and In-Vocabulary (IV) words. These error metrics are calculated in the same way with the WER, given in Equation 2.4, however, by only considering the substitution

and deletion errors. WERs for OOV words and IV words are formulated as follows:

$$WER_{OOV} \text{ (per cent)} = \frac{\#S_{OOV} + \#D_{OOV}}{\#OOV} \times 100 \quad (4.1)$$

$$WER_{IV} \text{ (per cent)} = \frac{\#S_{IV} + \#D_{IV}}{\#IV} \times 100 \quad (4.2)$$

Here $\#S_{OOV}$ and $\#S_{IV}$ respectively represent the number of substitution errors for OOV and IV words and $\#D_{OOV}$ and $\#D_{IV}$ respectively represent the number of deletion errors for OOV and IV words. $\#OOV$ and $\#IV$ also represent the number of OOV and IV words respectively. Considering these definitions the calculation of the WER is rewritten as

$$WER \text{ (per cent)} = \frac{\#S_{OOV} + \#D_{OOV} + \#S_{IV} + \#D_{IV} + \#I}{\#OOV + \#IV} \times 100 \quad (4.3)$$

where $\#I$ is the total number of insertion errors and $\#OOV + \#IV$ is equal to the number of reference words.

In our analysis, the reference words that do not occur in the vocabulary of the baseline word model (200 K vocabulary) are considered as OOV and the others are considered as IV words. Therefore, our analysis only investigates the gains obtained over the baseline word model. Note that the OOV error rate in the baseline word model is 100 per cent. The OOV and IV error analyses on the test set reveal the following.

1. All the sub-lexical approaches reduce the WERs for OOV words (from 100 per cent to 77 per cent for stem+ending and to 71 per cent on average for morph scenarios).
2. Morph scenarios that use the WB morph in the first-pass or rescoring degrade the accuracy of the IV words significantly.
3. Stem+ending and the best scoring morph scenario do not change the accuracy of the IV words.

Since the number of IV words is much larger than the number of OOV words, the performance deterioration of IV words is reflected in the overall WER (See Table 4.1).

We performed a more detailed OOV analysis for our best scoring model, the morph model. This analysis was performed on the automatic transcriptions of the linguistic segmentations since they yield better baselines than the acoustic segmentations. In sub-lexical approaches, word-like units are obtained from sub-lexical sequences by concatenating labeled units to the preceding units or by concatenating the units between word boundary markers to each other. These word like units may not be grammatically correct Turkish words all the time since sub-lexical sequences can also yield ungrammatical or non-word items. Therefore we call the concatenated sub-lexical sequences word-like units whether they are grammatically correct words or ungrammatical items.

In the morph-based model, first we analyzed the ratio of ungrammatical word-like units in the hypothesis sentences. The word-like units that do not occur in the 200 K word vocabulary were manually checked to determine if they are grammatically correct Turkish words or non-word items. We have found that 387 word-like units out of 22706 do not occur in the 200 K word vocabulary and 87 out of 387 are not grammatically correct Turkish words. These non-word items in the hypothesis sentences introduce 0.4 per cent WER to the morph model. This finding reveals that there is still room for further improvements on the morph-based model, i.e, by correcting the non-word items with a post-processing approach.

Second, we analyzed what percentage of the gain in the morph model is coming from handling the OOV words. The morph model was shown to outperform the word model by 1.0 per cent in Table 4.2. We have found that this 1.0 per cent gain is partitioned as (i) 0.6 per cent from recognizing OOV words correctly; (ii) 0.1 per cent from recognizing IV words correctly; (iii) 0.3 per cent from decreasing the number of insertions. Here it is important to note that, an OOV word introduces more than

one word error. 0.6 per cent of the gain out of 1.0 per cent is directly coming from dealing with the OOV problem using morphs as vocabulary units. The rest of the gain, obtained from improving the accuracy for IV words and reducing insertion errors, can be a consequence of correcting errors due to OOV words.

4.3.2. Low-level Analysis: Manual Classification of Recognition Errors

In this section, we perform a low level error analysis on the morph hypothesis sentences obtained on linguistic segmentations. The recognition errors can stem from several reasons such as OOV words, non-robust acoustic and language model estimates, search errors during decoding, etc. Automatic classification of recognition errors using acoustic and language model scores has been proposed in [108]. Our analysis is based on the manual classification of the recognition errors, similar to the error analysis given in [109], using only the raw ASR output. In this analysis, we do not consider any acoustic or language model information and the errors are classified according to some linguistic error types using the hypothesis sentences. The error types utilized in the classification are as follows:

1. **Proper name variations:** Some proper names are written slightly different in the language model training data and in the reference transcriptions. This variation is usually a single letter, however, it is counted as a word error.
   e.g., "Esad" in reference and *"Esat"* in hypothesis
2. **OOV words:** If the reference word is an OOV word, then the hypothesis word substituted with this OOV word is labeled as an error caused by OOV words.
3. **Similarity of reference and hypothesis words:** This error type evaluates the similarity of reference and hypothesis words according to acoustic similarity or suffix error. If the reference and the hypothesis words sound acoustically similar in a substitution error, this error is labeled as acoustic similarity error. If the reference and the hypothesis words have the same root but different suffixes, this error is labeled as suffix error.
   e.g., "ABD'li" in reference and *"Azeri"* in hypothesis, these two words sound similar.

e.g., "Haberlerden" in reference and *"Haberler"* in hypothesis, here *Haber* is the common root.

4. **Is the error correctable, incorrectable or linguistically correct?** The errors in the hypothesis sentences are labeled as correctable errors, incorrectable errors or linguistically correct errors. This subjective labeling is based on a simple idea. If we can correct the erroneous word without knowing the reference by simple syntactic and semantic changes on the hypothesis sentence, then this error is labeled as correctable. If we can not reach to the correct word with simple syntactic or semantic changes on the hypothesis sentence, then this error is labeled as incorrectable. Sometimes, there can be word errors in an hypothesis sentence, however, this sentence can be linguistically correct.
5. **Kind of errors:** The errors in the hypothesis sentences are labeled as being semantic errors or syntactic errors.
6. **Place of errors:** The errors in the hypothesis sentences are labeled as local level error or sentence level error. If an error is corrected by just looking at the adjacent words, then this error is labeled as a local level error. If we need to consider the whole sentence to correct the error, than this error is labeled as a sentence level error.

The examples for the last three error types will be explained using Figure 4.3. This figure shows three hypothesis sentences aligned with their reference transcriptions and annotated according to the given error classes. Reference transcriptions are denoted with "R" and the hypothesis sentences are denoted with "H". The erroneous words in the hypothesis sentences are written in capital letters. These words contain a set of associated labels given in curly braces. There are six label fields and the order of the labels is the same with the listed error types. As given in the third example in Figure 4.3, a group of consecutive erroneous words can be annotated together if they have the same error labels. In that case, the associated labels are attached to the last word of this group. If an error type is not applicable to the erroneous word, then this field is labeled with the ✘ sign. The acronyms utilized in the label fields are as follows:

```
R1:  hükümeti                              popülizm yapmakla suçladı
H1:  HÜKÜMETIN{ ✘, ✘, SE, CE, SYNE, SLE}  popülizm yapmakla suçladı

R2:  haberimizden                          izleyelim
H2:  HABERIMIZDE{✘, OOV, SE, GCE, ✘, ✘}   IZLEDI{✘, ✘, SE, GCE, ✘, ✘}

R3:  saatler washington'da                          yirmi otuz
H3:  saatler AŞIRET ON BEŞ{✘, ✘, ICE, SEME, SLE}  yirmi otuz
```

Figure 4.3. Hypothesis (H) sentences aligned with their reference (R) transcriptions and annotated according to the given error classes.

PNV	proper name variation error
OOV	OOV word error
ASE	acoustic similarity error
SE	suffix error
CE	correctable error
ICE	incorrectable error
LCE	linguistically correct error
SEME	semantic error
SYNE	syntactic error
LLE	local level error
SLE	sentence level error

In the first example, there is no proper name variation and the error is not due to an OOV word since these fields are labeled with the ✘ sign. The reference and the hypothesis words have the same root but different suffixes, therefore the error is labeled as a suffix error. Any native Turkish speaker can understand that this hypothesis sentence has an ungrammatical structure due to the incorrect suffix in the first word. The verb of this sentence conveys the syntactic information for suffix correction. Therefore this syntactic and sentence level error is labelled as a correctable error. In the second example, an error has occurred in the first word due to an OOV word, "haberimizden", however, the root of this word is recognized correctly. Another suffix error has occurred in the second erroneous word. Even though, both of the words are

recognized incorrectly, the hypothesis sentence is linguistically correct. Linguistically correct hypothesis sentence means that, the sentence sounds linguistically correct even though there are word errors. In the third example, the hypothesis sentence has a nonsense meaning and it is impossible to reach the reference words from hypothesis errors without knowing the reference. Therefore, the erroneous words are labelled as incorrectable. The errors in this hypothesis are semantic errors at the sentence level.

We performed the error analysis on the hypothesis sentences of the baseline morph model. We only annotated the sentences containing approximately less than 35 per cent erroneous words to make accurate decisions about the type of word errors. We did not label the erroneous words if we were not certain about the error types. The labelled words only contain 6.4 per cent of the recognition errors out of 22.4%. We have found that

- 2.6 per cent out of 6.4 per cent word errors are labelled as acoustic similarity errors. Among this 2.6 per cent word errors, 0.1 per cent are labelled as correctable, 1.6 per cent are labelled as incorrectable and 0.9 per cent are labelled as linguistically correct.
- 2.5 per cent out of 6.4 per cent word errors are labelled as suffix errors. Among this 2.5 per cent word errors, 1.3 per cent are labelled as correctable, 0.1 per cent are labelled as incorrectable and 1.1 per cent are labelled as linguistically correct.
- 1.3 per cent out of 6.4 per cent word errors are categorized neither as acoustic similarity nor suffix errors.

The conclusion that can be reached from these analyses is that if a word error is caused due to an incorrect suffix in the hypothesis word, there is almost a 45 per cent possibility of correcting this error with a post processing approach on the raw ASR output.

Then we investigated the percentage of the erroneous words recognized correctly in the 1000-best oracle. Here the motivation is that if the erroneous words are recognized correctly in the oracle, then there is a possibility of correcting these errors with reranking approaches. We have found that 4.0 per cent of the labelled word errors

out of 6.4 per cent are correctly recognized in the 1000-best oracle hypotheses. Our subjective labelling on this 4.0 per cent word errors reveals that

- 1.2 per cent out of 4.0 per cent word errors are labelled as correctable and 99 per cent of the correctable errors are due to syntactic errors.
- 1.1 per cent out of 4.0 per cent word errors are labelled as incorrectable and 58 per cent of the incorrectable errors are due to semantic errors.
- 1.7 per cent out of 4.0 per cent word errors are labelled as linguistically correct.

The conclusion that can be reached from these analyses is that the main cause of the word errors labelled as correctable is the incorrect syntax. Regarding this finding, we can say that there is a possibility of correcting almost 1.2 per cent of the recognition errors with reranking approaches that address the syntactic errors. This leads us to explore discriminative language models with syntactic features in Chapter 6.

Additionally, linguistically correct errors have the highest percentage in the partitioning of the 4.0 per cent word errors. There is no way of dealing with these errors using only the raw ASR output. Therefore, complementary information sources, i.e., acoustic model scores, need to be taken into account for these errors.

5. LATTICE EXTENSION AND VOCABULARY ADAPTATION

This chapter focuses on handling the OOV problem in word-based Turkish ASR. OOV words are considered as one of the significant sources of recognition errors. An OOV word is exchanged with an IV word during decoding and introduces on average 1.5 recognition errors [19]. Therefore, dealing with the OOV problem is crucial for the ASR systems suffering from high OOV rates.

A commonly proposed solution to large number of OOV words in agglutinative languages is to use sub-lexical recognition units instead of words as explained in Chapter 4. We found out that sub-lexical units solve the OOV problem and outperform words in the recognition accuracy for Turkish even for the word-based system yielding 1 per cent OOV rate. Additionally, we observed that the performance of the word-based system is getting closer to the performance of the best sub-lexical system with the increasing vocabulary size (See Table 4.1). However, very large vocabulary sizes require more memory and computational power. In addition, including rare words in the vocabulary results in non-robust language model estimates due to data sparsity, as a result, very large vocabularies may degrade the system performance.

Table 3.3 reports the results of a cheating experiment where full vocabulary coverage is achieved without increasing the vocabulary size drastically. This cheating experiment achieves the same performance with the best sub-lexical approach. This finding leads us to deal with the OOV problem directly on the word-based system with a similar approach to this cheating experiment. Instead of increasing the vocabulary size using the most frequent words in the text data, we dynamically adapt the baseline vocabulary using a multi-pass recognition strategy. Our proposed approaches increase the adapted vocabulary size compared to the baseline vocabulary to an extent where one can still benefit from higher coverage without sacrificing robustness. In addition to the limited vocabulary sizes, OOV words can be introduced to the ASR systems due to

the mismatch between the topic of the test utterances and the recognition vocabulary. In order to handle this mismatch, our approaches consider the first-pass ASR output of the baseline system as the prior knowledge to adapt the recognition vocabulary for the second-pass. The new words that will be added to the baseline vocabulary are learned using the baseline recognition output with the assumption that OOV words are replaced by acoustically *similar* IV words during decoding.

In this research, we explore lattice extension and vocabulary adaptation techniques to handle the OOV problem in a word-based ASR system with a moderate vocabulary size, 50 K. These multi-pass vocabulary adaptation techniques were formerly proposed in [55] and [50] respectively. In this research we investigate the effectiveness of these approaches on the Turkish LVCSR system and compare their performance with larger predetermined vocabularies utilized in the first-pass recognition. Predetermined vocabularies are generated with the most frequent words in the text data related to the ASR application domain in advance of decoding. As was mentioned in Section 2.3.2, a few similarity criteria have been defined for lattice extension and vocabulary adaptation techniques to find the similar words that will be added to the adapted vocabulary. As one of the main contributions of this research, we introduce a new similarity criterion that takes the agglutinative language characteristics of Turkish into account. This similarity criterion is called position-dependent phonetic-distance similarity.

As another contribution of this research, lattice extension technique is also introduced for statistically derived sub-lexical units, morphs, to handle non-word recognition sequences in the morph recognition outputs. On one hand, using sub-lexical units alleviates the OOV problem, on the other hand sub-lexical units may result in ungrammatical items since they can generate any combination of sub-lexical units which include non-word items. Over-generated units can be corrected with simple morphological constraints if the sub-lexical units convey morphological features [87]. Since statistical morphs do not carry explicit linguistic information, the lattice extension strategy is modified to map morph sequences to grammatically correct Turkish words. The following sections will explain the details of lattice extension both for words and

morphs and vocabulary adaptation only for words. The newspaper content transcription system, explained in Section 3.3.1, is utilized in the experiments.

5.1. Methods

In this section, we will first describe the details of lattice extension and vocabulary adaptation. Then the similarity criteria utilized by these techniques will be explained thoroughly. The following definitions will be used in the explanation of the methods.

V_s	Recognition vocabulary: $w_s \in V_s$ and $\|V_s\| = 50$K
V_f	Fallback vocabulary: $w_f \in V_f$, $\|V_f\| = 675$K and $V_s \subseteq V_f$
V_m	Morph vocabulary: $m \in V_m$ and $\|V_m\| = 34.3K$
V_{seq}	Morph sequence vocabulary: $m_{seq} \in V_{seq}$ where $V_{seq} \subseteq V_m^+$
$D(\cdot,\cdot)$	Pairwise string distance function: A function which assigns a weight to each pair of strings, (x, y), where $x \in V_s$ or $x \in V_{seq}$ and $y \in V_f$.
$r(\cdot)$	Root/First-morph function: A function which returns the root of a word or the first-morph of a morph sequence corresponding to a word.

5.1.1. Lattice Extension

For agglutinative languages, high OOV rates is one of the main problems of the word-based recognition systems with moderate size vocabularies. Lattice extension aims to alleviate this problem by modifying the recognition output with *similar words* from a larger vocabulary which may include the existing OOV words. In our experiments, a fallback vocabulary (V_f) which contains all the word types, 675 K words, in the training corpus (Text-I corpus, see Section 3.1.2 for the details) is used as the larger vocabulary. This vocabulary contains 97.5 per cent of the word types and 98.5 per cent of the word tokens in the test data.

As was shown in Chapter 4, the statistical sub-lexical approach, morphs, yields a better recognition accuracy than words. However, since any combination of the morphs can be generated by the decoder, recognition output can contain ungrammatical items.

When word-like units are obtained from hypothesized morph sequences, we have found out that 387 word-like units out of 22706 do not occur in the 200 K word vocabulary and 87 out of 387 are not grammatically correct Turkish words. The most common errors are the wrong detection of word boundary, incorrect morphotactics and totally meaningless sequences that are acoustically similar to the reference words. These non-word items in the hypothesis sentences introduced 0.4 per cent WER to the morph model in Chapter 4. Therefore, recognition accuracy of morphs can be improved if over-generated items are rectified. Simple morphological constraints can be applied as a post processing approach to handle the ungrammatical words as suggested in [87]. However, morphs are data driven units and unlike grammatical morphemes they do not provide explicit morphological features. Therefore, we adapt the lattice extension strategy to correct the over-generated items in morphs.

The main steps in lattice extension are given in the flow chart in Figure 5.1. The hypothesis word lattice is generated by decoding the acoustic segmentations with the baseline acoustic and language models. Note that the baseline language model was built with the recognition vocabulary (V_s) words. An example lattice output is shown in Figure 5.2. In the original lattice, there should be acoustic and language model costs, however these costs are removed from the lattice for our approach since there is no need to use them in lattice extension. Every word in the hypothesis lattice is extended with *similar words* from V_f and the new lattice is called the extended lattice (Figure 5.4). This process is implemented with FSTs. The dashed box in the figure contains the main blocks required for generating the extended lattice from the hypothesis lattice. First, the pairwise string distance functions between each word in the recognition vocabulary, w_s, and each word in the fallback vocabulary, w_f, are calculated. This function is denoted by $D(w_s, w_f)$. Second, a single state costless transducer ($w_s 2 w_f$) is built using $D(w_s, w_f)$. The transducer includes an arc labeled w_s:w_f if $D(w_s, w_f)$ is less than a given threshold τ (Figure 5.3). Third, lattice output is composed with the $w_s 2 w_f$ transducer and the arc symbols in the composition output are projected to w_f to obtain the extended lattice (Figure 5.4).

After generating the extended lattice, a second-pass recognition is performed on

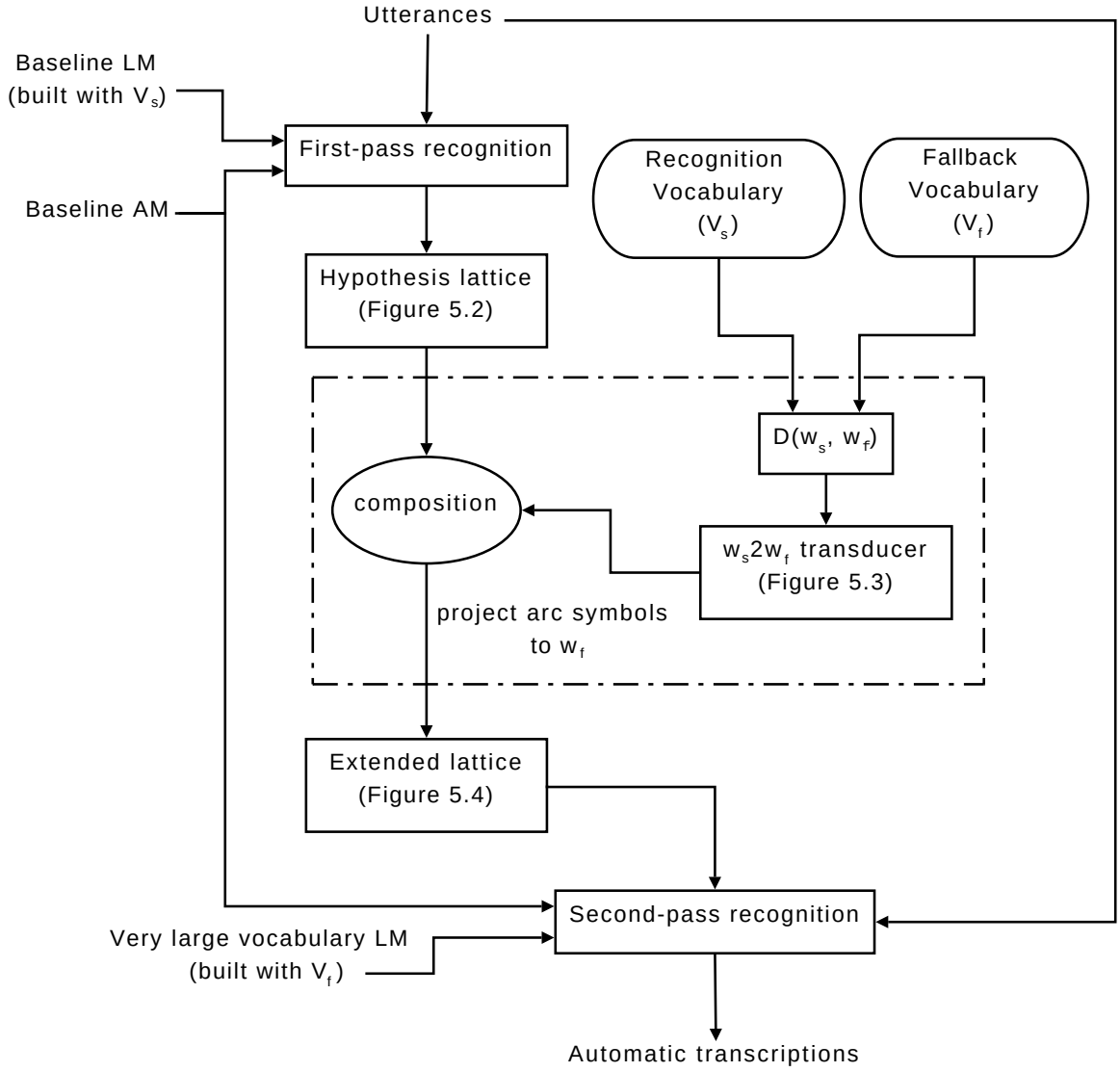


Figure 5.1. The flowchart showing the main steps in lattice extension for words.

the extended lattice. Since no contextual information is used during the extension of each word, a larger vocabulary language model, built with V_f, is also utilized in the second-pass recognition. Note that the extended lattice can be over-generated, however, the larger vocabulary language model helps the decoder to eliminate syntactically and semantically incorrect sentences. In [55], acoustic segmentation specific language models were used for the second-pass recognition of each extended utterance lattice. These language models were built with adapted vocabularies generated by replacing the least frequent words of the baseline vocabulary with the new words in the extended lattice. This approach has the advantage of limiting the vocabulary size after adaptation, however, may result in expensive language model computations for each utterance. In

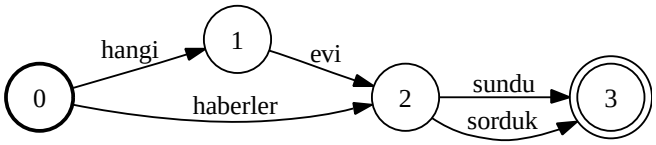


Figure 5.2. Lattice output of the baseline word-based recognizer.

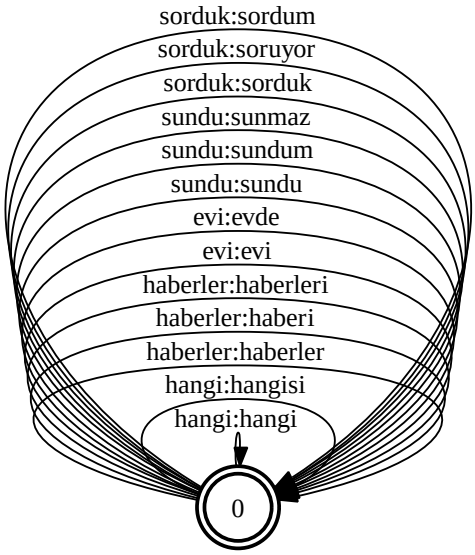


Figure 5.3. $w_s 2 w_f$ transducer.

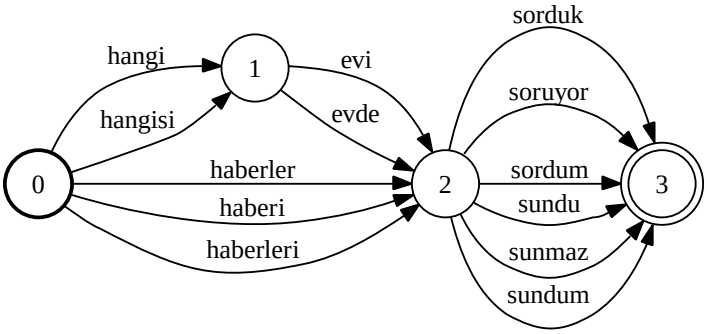


Figure 5.4. Extended lattice, generated by composing the hypothesis lattice with the $w_s 2 w_f$ transducer. Arc symbols are projected to w_f.

order to reduce expensive computations, we used a single large vocabulary language model built with V_f in our experiments. The disadvantage of our approach is that large vocabulary language models may suffer from non-robust estimates due to data sparseness.

In lattice extension for morphs, the morph lattice is converted to a morph sequence lattice with the help of the word boundary markers. Unlike the lattice given in Figure 5.2, the morph sequence lattice may contain ungrammatical items. Note that concatenation of morphs may result in invalid words. Then this lattice is composed with a single state costless transducer ($m_{seq}2w_f$) built using $D(w_{seq}, w_f)$ and the arc symbols in the composition output are projected to w_f to generate the extended lattice. By this way an extended word lattice containing only the words from V_f is generated from the baseline morph lattice. Then second-pass recognition is performed with this lattice and a larger vocabulary language model.

The main difference between word and morph lattice extension experiments can be explained as follows using the flowchart given in Figure 5.1. In morph lattice extension,

- first-pass recognition is performed with the baseline morph language model instead of the language model built with V_s,
- the first-pass recognition output contains the morph lattice and there needs to be another block between first-pass recognition and hypothesis lattice blocks to represent the conversion of morph lattice to morph sequence (word-like units) lattice,
- the recognition vocabulary, V_s, block needs to be replaced with morph sequence vocabulary, V_{mseq}, for generating the $m_{seq}2w_f$ transducer.

5.1.2. Vocabulary Adaptation

Vocabulary adaptation is applied to the word-based model to handle the OOV problem. Since a mismatch between the vocabulary and the test data will result in OOV

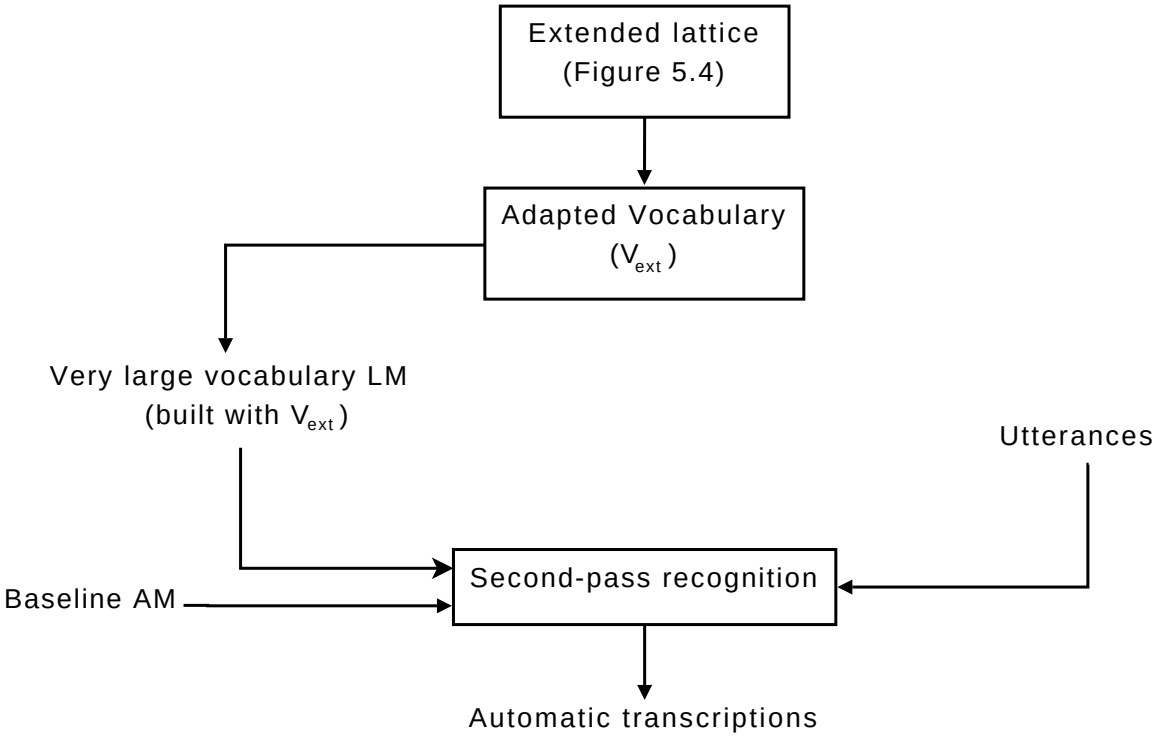


Figure 5.5. The flowchart showing the main steps in vocabulary adaptation for words.

words, the main idea in vocabulary adaptation is to alter the recognition vocabulary with a prior knowledge of the test data. In this approach, the hypothesis lattice is used as the prior information for vocabulary adaptation. The adapted vocabulary is generated from V_f words that are similar to the lattice word types.

Figure 5.5 summarizes the main steps in vocabulary adaptation. In order to find the similar words to hypothesis lattice words, we generate the extended lattice as explained in Section 5.1.1. The word types in the extended lattice are used as the adapted vocabulary (V_{ext}, where $V_{ext} \subseteq V_f$). Same similarity criteria with the lattice extension are used to modify the vocabulary. More formally, $V_{ext} = \{w : w \in V_f, \exists w_s \in V_{lat}, D(w_s, w) < \tau\}$ for some distance function $D(w_s, w)$ and threshold τ.

The main difference between the lattice extension and vocabulary adaptation techniques is in the second-pass recognition. After an extended lattice is generated, the word types in the extended lattice are used as the altered vocabulary and a new language model is built with this larger and adapted vocabulary, V_{ext}. In lattice extension, second-pass recognition is performed with the larger vocabulary language model

and the extended lattice where the search space of the decoder is constrained with the extended lattice (See Figure 5.1). However, in vocabulary adaptation, second-pass recognition is performed only with the larger vocabulary language model.

5.1.3. Similarity Criteria

In this chapter, we are only interested in handling the errors due to OOV words. An OOV word will most probably be replaced with an IV word during decoding. We assume that this replacement is between similar word pairs. Therefore, several similarity measures based on pairwise distance functions are defined to find the feasible replacements between IV and OOV words.

The similarity criteria used in this study are classified as morphology-based, first-morph-based and phonetic-distance-based. These criteria take the characteristics of Turkish into account while defining the similarity measures. In this section, these similarity measures will be explained explicitly.

5.1.3.1. Morphology-based Similarity. For agglutinative languages like Turkish, many new word forms can be derived from a single stem (See Figure 2.5). Although a word may not exist in V_s, the root of that word can occur in the same vocabulary with another inflectional or derivational form. In our baseline newspaper content transcription system, 59.2 per cent of the root types of the OOV words exist in V_s with different word forms. In addition, it was stated in [55] that for Serbo-Croatian "a large number of words in the recognized hypothesis are recognized incorrectly because only the inflection ending is wrong whereas the stem was recognized correctly". Therefore, correcting the errors caused by replacing an OOV word with an IV word *having the same root* is the main motivation for this similarity criterion.

In our implementation, the roots were obtained using a simple stemmer instead of a Turkish morphological analyzer [18, 91, 92], since we were not interested in detailed morpheme decompositions. The stemmer was implemented as a weighted FST using

a given list of 35.6 K roots and 392 surface form suffixes. This stemmer can generate the roots of the words without taking morphotactics into account. Words that could not be processed by the stemmer remain untouched. Over-stemming was handled by assigning costs to the suffixes and selecting the root with the lowest cost. This yielded the longest root among all possible candidates.

In morphology-based similarity criterion, a binary distance function is defined as follows and the $w_s 2 w_f$ transducer only includes pairs with distance 0.

$$D(w_s, w_f) = \begin{cases} 0 & \text{if } r(w_s) = r(w_f) \\ 1 & \text{otherwise} \end{cases} \tag{5.1}$$

Briefly, w_s can be extended with any fallback vocabulary word having the same root. The average number of words for a common root was calculated as 169.8 in the fallback vocabulary. This was also equal to the average number of mappings from lattice word tokens to fallback vocabulary words in the $D(w_s, w_f)$ transducer. Maximum number of mappings was found to be 1662 for the root "yap *(to make)*".

5.1.3.2. First-morph-based Similarity. This similarity is proposed to eliminate the over-generated units in the sub-word approach by mapping the morph sequences to grammatically correct words.

The algorithm for generating morphs does not aim to find the correct morphological analysis. The principal idea in morphs is to find a concise set of language modeling units relevant for speech recognition using a data-driven approach. Thus, statistical morphs do not provide explicit linguistic information like grammatical morphemes. In this similarity criterion, we assume an analogy between the roots and the first morphs since Turkish is an agglutinative language where all the affixes are suffixes. The first morph of each word is considered as the most semantic information bearing part of that word like roots and morphology-based similarity measures are modified using the

first-morph-based similarity. The binary distance function is defined as:

$$D(m_{seq}, w_f) = \begin{cases} 0 & \text{if } r(m_{seq}) = r(w_f) \\ 1 & \text{otherwise} \end{cases} \tag{5.2}$$

The $m_{seq}2w_f$ transducer only includes pairs with distance 0.

5.1.3.3. Phonetic distance-based Similarity. Different than the morphology-based similarity, the assumption in this criterion is that an OOV word can be replaced with any IV word that is close in the acoustic space. The level of closeness is measured using the Levenshtein distance, the minimum number of edit (substitution, deletion, insertion) operations needed to convert one string into another. In our system, phonetic distance is the same as grapheme distance since the acoustic models are based on letters instead phonemes.

If x^T denotes an arbitrary string of length T, and x_j denotes the letter at the j'th position, the Minimum Edit Distance (MED) between two strings $d(x^t, y^v)$ is defined as

$$d_{MED}(x^t, y^v) = \min \left\{ \begin{array}{c} c_t(x_t, y_v) + d_{MED}(x^{t-1}, y^{v-1}), \\ c_t(x_t, \epsilon) + d_{MED}(x^{t-1}, y^v), \\ c_t(\epsilon, y_v) + d_{MED}(x^t, y^{v-1}) \end{array} \right\} \tag{5.3}$$

where $c_t(x_t, y_v)$, $c_t(x_t, \epsilon)$, $c_t(\epsilon, y_v)$ are respectively the substitution, deletion and insertion costs at the t'th position of the reference string, and

$$c_t(x_t, \epsilon) = c_t(\epsilon, y_v) = 1 \text{ and } c_t(x_t, y_v) = \begin{cases} 1 \text{ if } x_t \neq y_v, \\ 0 \text{ if } x_t = y_v. \end{cases} \tag{5.4}$$

We modified MED using position dependent costs for each edit operation and call this Position Dependent Minimum Edit Distance (PDMED). Since in Turkish all affixes are suffixes, we thought that two strings are more similar if the edit operations

occur at the end of the strings. Therefore, PDMED penalizes the edit operations at the beginning of the strings more than the operations at the end of the strings. In this implementation the cost of edit operations are considered as a function of the reference string length (T) and the position of the edit operation (t) in the reference string. So, in this case,

$$c_t(x_t, y_v) = c_t(x_t, \epsilon) = c_t(\epsilon, y_v) = f(t, T) = 2 - \frac{2 * t}{T + 1} \tag{5.5}$$

where, $0 < f(t, T) < 2$. Although, the minimum edit distances of two string pairs (abc, dbc) and (abc, abd) are equal, PDMED assigns a larger distance to the first string pair. The edit distances between string pairs also depend on the lengths of the strings. Therefore, the distances are normalized with the length of the reference string ($|x^t| = t$) to compare them with the same threshold.

Phonetic distance-based similarity was both applied to word and morph lattices with different distance functions. During the generation of the extended lattices, following distance functions were calculated.

1. MED-words: Word similarity with MED
   $D(w_s, w_f) = d_{MED}(w_s, w_f)/|w_s|$
2. PDMED-words: Word similarity with PDMED
   $D(w_s, w_f) = d_{PDMED}(w_s, w_f)/|w_s|$
3. MED-roots: Root similarity with MED
   $D(w_s, w_f) = d_{MED}(r(w_s), r(w_f))/|r(w_s)|$
4. PDMED-roots: Root similarity with PDMED
   $D(w_s, w_f) = d_{PDMED}(r(w_s), r(w_f))/|r(w_s)|$
5. MED-morphs: Similarity of morph sequences and words with MED
   $D(m_{seq}, w_f) = d_{MED}(m_{seq}, w_f)/|m_{seq}|$

where $r(w_s)$ and $r(w_f)$ are the roots of the words w_s and w_f respectively and m_{seq} is the morph sequence corresponding to a word generated using the word boundary information. In the first two distance functions the costs are calculated between the

given words. However, in the third and the fourth functions, the distance between words are computed as the distance between corresponding roots. The motivation behind these distance functions is to take the errors in the root of the words into account. We also applied phonetic-distance-based similarity to morphs to find feasible mappings from over-generated morph sequences to grammatically correct Turkish words. The w_s2w_f and $m_{seq}2w_f$ transducers only include pairs whose distance is less than a given threshold. We experimented with different thresholds to explore the effect of extended vocabulary size on the recognition performance.

5.2. Results

In this section, experimental results are given for word lattice extension, vocabulary adaptation and morph lattice extension approaches using different similarity criteria. We also conducted larger vocabulary single-pass recognition experiments with predetermined vocabularies and compare their results with the proposed techniques.

5.2.1. Baseline ASR Systems

We utilized the newspaper content transcription system in the experiments. The lattice extension and vocabulary adaptation experiments for words were performed on the baseline word system that yields 11.8 per cent OOV rate with 38.8 per cent WER. The details of this system were explained in Section 3.3.1. For the morph lattice extension experiments, we also utilized the newspaper content transcription system with the morph vocabulary and the morph language model. The morph vocabulary was learned on the word types of the Text-I corpus using the baseline Morfessor algorithm. For robustness, only the word types occurring at least 2 times were used, resulting in 34.3 K morphs. Remaining word types were segmented into morphs with the Viterbi algorithm using the initial segmentations. In order to facilitate converting morph sequences into word sequences, the word boundaries were marked with a special morph, #, and the ratio of morph tokens to word tokens was calculated as 2.4 including the word boundary symbol. Note that the word boundary detection scenario with marking the non-initial morphs resulted in better results than using the word boundary morph

in Chapter 4. However marking the non-initial morphs in the newspaper transcription system increased the vocabulary size from 34.3 K to 47 K which is closer to the word vocabulary size, 50 K. Since morph model suggests higher order n-grams than the word model, we did not experiment with 47 K morph vocabulary not to introduce data sparseness problem to the morph model. Note that the newspaper content transcription system utilizes approximately one seventh of the training text data used in the BN transcription system. The best results were obtained using 3-gram word and 5-gram morph language models. The baseline results are summarized in Table 5.1.

Table 5.1. Results for the baseline systems

Experiments	Lexicon (x10^3)	Test Coverage (per cent)	WER (per cent)
Baseline-word	50	88.2	38.8
Baseline-morph	34.3	100	33.9

5.2.2. Lattice Extension Experiments with Words

In word lattice extension experiments, morphology-based and phonetic-distance-based similarity criteria were used to find the *similar words* from V_f. Figure 5.6 compares the average number of mappings from hypothesis lattice word tokens to fallback vocabulary words with respect to the distance threshold for all the similarity criteria. The size of the extended lattice is directly proportional with the average number of mappings for each token. Therefore, the average number of mappings and the extended lattice size will be used to refer to the same concept. The average number of mappings are higher for PDMED than classical edit distance with the same threshold since PDMED lightly penalizes the edit operations at the end of the strings. In our experiments, we aimed to keep the average number of mappings for each criterion approximately equal for a fair comparison of the similarity measures. Therefore, distance thresholds were set to give the same average number of mappings for each criterion. We experimented at several average number of mapping points to see the effect of extended lattice size on the recognition performance.

In lattice extension experiments, the quality of the extended lattices were com-

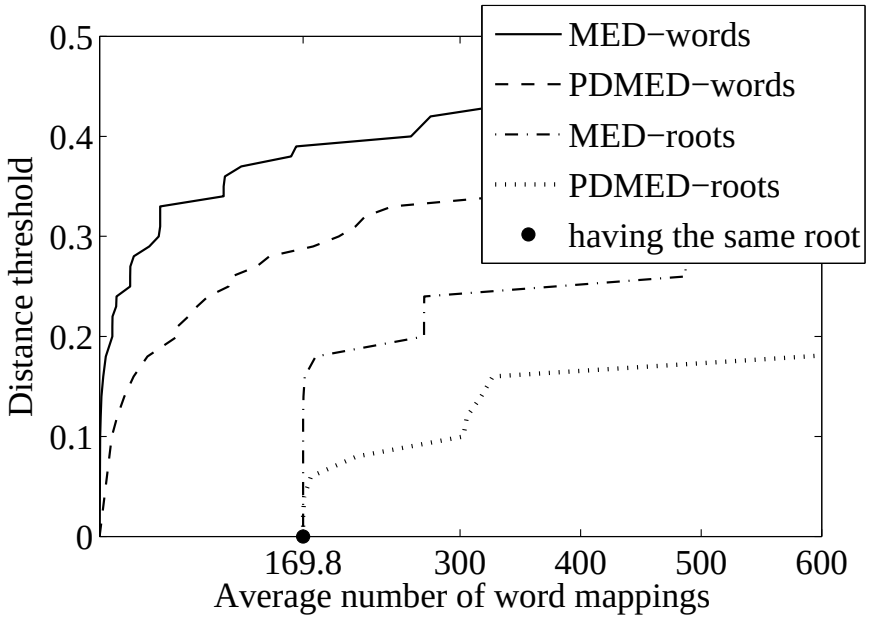


Figure 5.6. The average number of mappings from hypothesis lattice word tokens to fallback vocabulary words for different distance thresholds. Distance threshold of "0" corresponds to the original lattice.

pared in terms of Lattice Word Error Rate (LWER), in other words oracle WER, for different extended lattice sizes. Figure 5.7 shows the relationship between the LWER and average number of mappings for each similarity criteria. As expected, increasing the average number of mappings for hypothesis lattice words decreases the LWER. However, for the same average number of mappings, phonetic-distance-based similarity between words gives lower LWERs, since phonetic-distance-based similarity between roots drastically extends every word in the hypothesis lattice. In addition, classical and position dependent minimum edit distance do not reveal any significant difference.

The word error rate results are given in Figure 5.8. As seen in the figure, there is a linear relationship between the extended lattice LWERs and the second-pass recognition WERs. Morphology-based similarity criterion, *having the same root*, gives higher WER than phonetic-distance-based similarity for the same LWER. We obtained our best result as 34.2 per cent with MED-words similarity criterion and this yielded 4.6 per cent absolute improvement over the baseline. Even though the baseline morph model is 0.3 per cent better than our best word lattice extension result, this difference is not statistically significant. This lead us to conclude that lattice extension with a

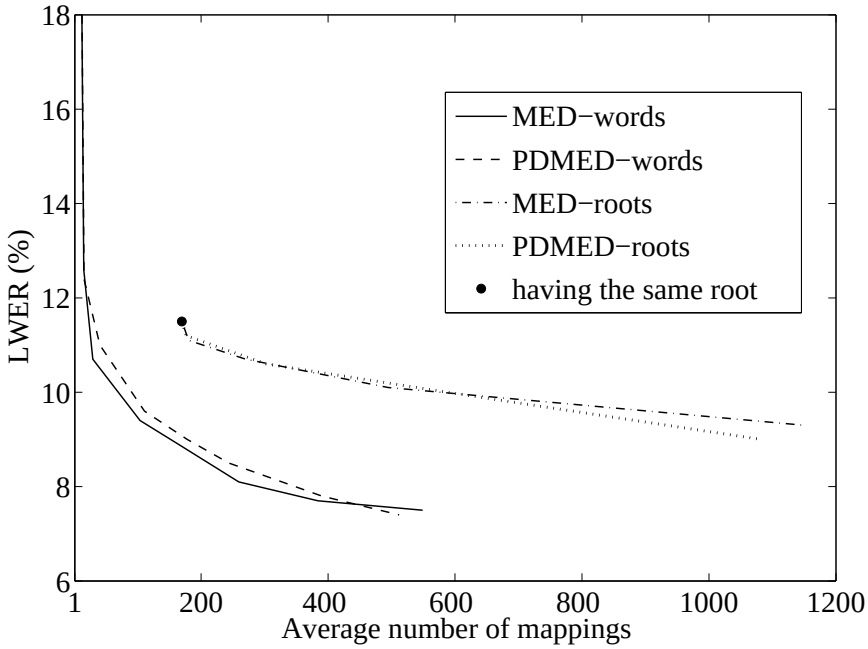


Figure 5.7. Effect of average number of mappings on LWER.

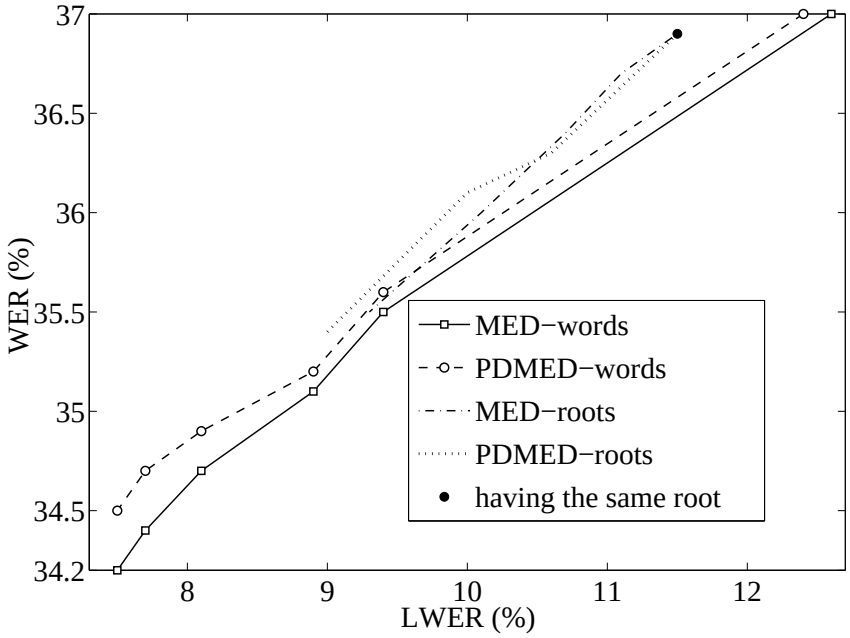


Figure 5.8. Relationship between LWER and recognition accuracy.

huge vocabulary language model can alleviate the OOV problem when a moderate vocabulary size is used for the first-pass recognition. However, sub-word units achieve a similar performance with a much smaller vocabulary size at the first-pass recognition.

In addition to using distance thresholds for MED-words, we extended the lattice

with the most similar N words where N varies from 5 to 500. We observed that increasing N also decreases the WER but no significant difference between this approach and using distance thresholds was seen in the recognition performance for similar size extended lattices.

5.2.3. Vocabulary Adaptation Experiments with Words

In vocabulary adaptation, the baseline vocabulary was modified by adding all the word types in the extended lattice. The size of the adapted vocabulary increases with increasing average number of mappings and larger vocabularies require excessive language model pruning. Therefore, the recognition output was rescored with an unpruned language model built with the same vocabulary to reduce the effect of excessive pruning[14]. In the adaptation experiments, the effect of the adapted vocabulary size on the recognition performance was investigated and the results were compared with the lattice extension experiments. In addition, recognition experiments were also performed using predetermined vocabularies containing the words that are most frequently seen in the training text.

Figure 5.9 compares the vocabularies generated from the extended word lattices in terms of test OOV rate for different extended lattice sizes. OOV rates are also given for the same size predetermined vocabularies. Although the effect of increasing the vocabulary size is noticeable in all the similarity techniques, predetermined vocabularies give better coverage compared to adapting vocabularies with extended lattices.

Recognition experiments were only performed for the adapted vocabularies generated with $d_{MED}(w_s, w_f)$ distance function. The WER results for different size vocabularies are shown in Figure 5.10. This figure reveals some interesting points. First, vocabulary adaptation performs better than lattice extension for the same extended lattices[15]. For our largest vocabulary size, 591 K words, both of the approaches give

[14] In the lattice extension experiments, second-pass recognition was performed with unpruned language models since larger language models can be handled by constraining the search space of the decoder with extended lattice.

[15] In the lattice extension, a single language model built with V_f was used for all the experiments.

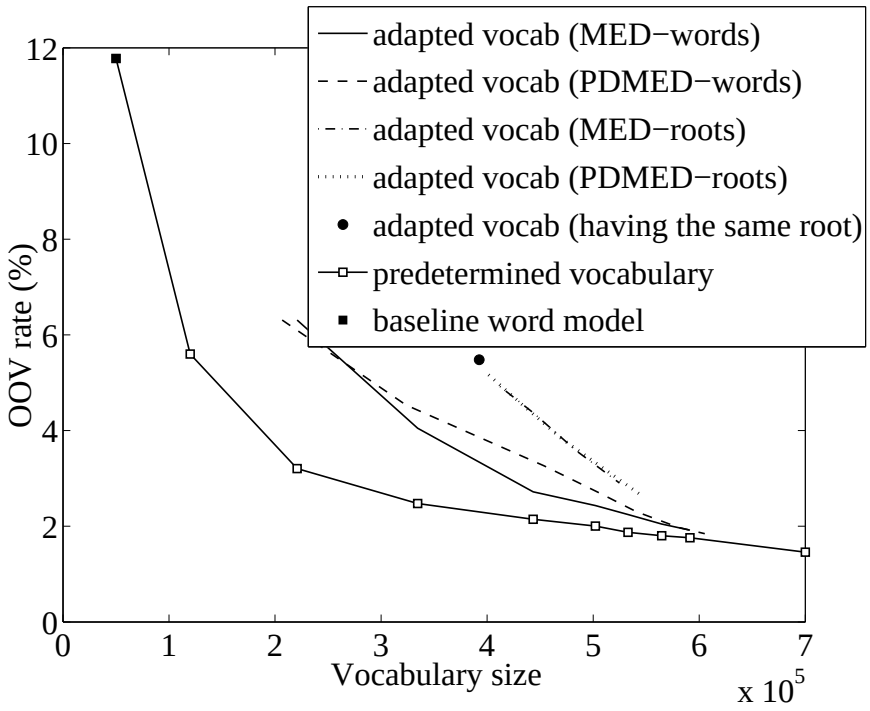


Figure 5.9. Effect of vocabulary size on OOV rate for different similarities.

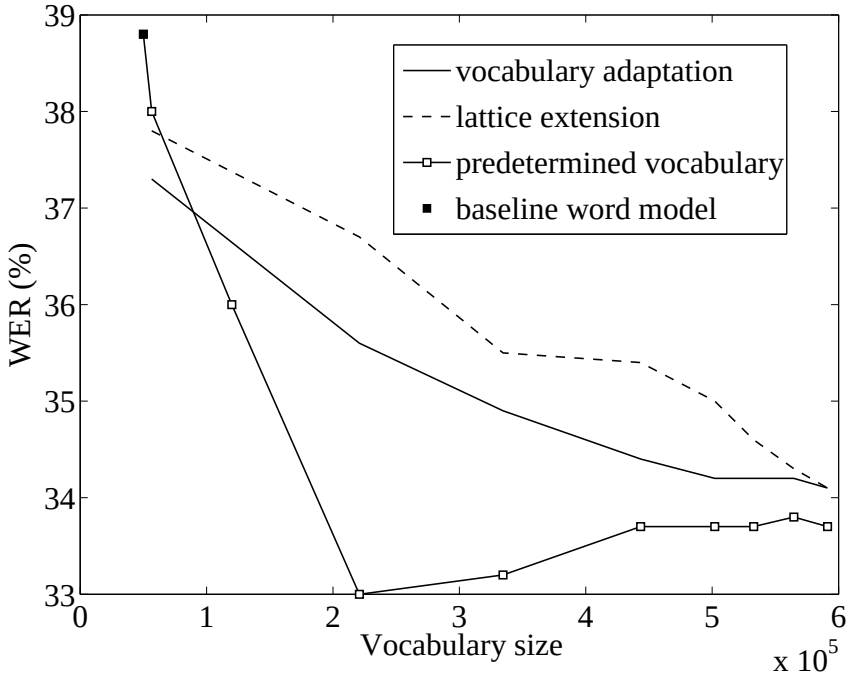


Figure 5.10. Effect of vocabulary size on WER for MED-words similarity.

the same result. Second, recognition with predetermined vocabularies achieves bet-

Therefore, we repeated the MED-words lattice extension experiments using unpruned extended lattice specific language models to compare their results with vocabulary adaptation approach.

Table 5.2. Results for morph lattice extension experiments

Experiment	Average number of mappings	WER (per cent)
Baseline morph model	1.0	33.9
Morph-based similarity	21.15	32.8
MED-morphs	22.37	32.4
MED-morphs-Nwords	25	32.3

ter recognition accuracy than vocabulary adaptation and lattice extension approaches. This can be explained with the lower OOV rates for the predetermined vocabularies. Third, best WER is attained with 221 K predetermined vocabulary and then recognition accuracy starts to decrease with the increasing vocabulary size. This can be explained with excessive language model pruning at the first-pass recognition and the non-robust language model estimates due to the larger vocabularies.

5.2.4. Lattice Extension Experiments with Morphs

In morph lattice extension experiments, first-morph-based similarity and phonetic-distance-based similarity were used to handle the ungrammatical items. We only utilized $d_{MED}(m_{seq}, w_f)$ as the distance function in phonetic-distance-based similarity. We conducted trials at different similarity distance thresholds. In addition, each token in the hypothesis lattice was also extended with the most similar N words from V_f where N varies from 5 to 50 (called MED-morphs-Nwords). Table 5.2 summarizes the comparison of the similarity criteria used in morph lattice extension experiments.

Although lattice extension for morphs brings back the OOV problem, 1.1 – 1.6 per cent absolute accuracy improvements are achieved with the proposed similarity criteria. Similar to word lattice extension experiments, phonetic-distance-based similarity performs better than first-morph similarity. The best result over the baseline morph model is obtained when each morph sequence in the hypothesis lattice was extended with 25 most similar words using MED similarity criteria. This improvement is statistically significant at p=0.002 as measured by the NIST MAPSSWE significance test.

This is also better than the result obtained with the 221 K predetermined vocabulary.

5.3. Analysis and Discussion

In this Chapter, word lattice extension and vocabulary adaptation techniques were applied to the lattice output of the baseline word system. The baseline 50 K word vocabulary resulted in 11.8 per cent OOV rate and yielded 38.8 per cent WER. Although it is practically and theoretically impossible to add all the available words to the lexicon, a cheating experiment with 50 K vocabulary where the infrequent words from the original vocabulary was replaced with the OOV words got the WER down to 30.1 per cent. This is our lower bound for the 50 K word-based system. We also performed the same cheating experiment with 221 K vocabulary, which gives the best result in Figure 5.10, and obtained 31.5 per cent WER. It is important to note that, due to the inclusion of rare words in the language model, 221 K vocabulary resulted in a higher WER compared to 50 K vocabulary when there were no OOV words.

In word lattice extension experiments, the MED and the modified version of MED (PDMED) gave almost the same results. The best result was obtained as 34.2 per cent. The main difference between lattice extension and vocabulary adaptation is in the search space of the second-pass recognition. Due to the constrained search space, lattice extension fails in finding the correct word in the second-pass recognition, if the OOV word is not added to the correct position on the extended lattice. In order to evaluate the performance of this approach when all the reference words, both IV and OOV, are added to the correct position on the extended lattice, we performed a simple cheating experiment. A sausage like lattice was generated by adding new arcs to the one best hypothesis for the reference words corresponding to the substitution and deletion errors. Then this lattice was extended with similar words from a fallback vocabulary containing only the word types in the first-pass hypothesis lattice and the reference words (29 K words). The aim of using a smaller fallback vocabulary is to eliminate the data sparseness problem in the second-pass recognition. This experiment reduced the WER to 28.3 per cent. This is our lower WER bound for the lattice extension scenario.

Table 5.3. Results for extending the one best hypothesis of 50 K system

Experiment	Vocabulary	Adapted vocabulary	WER (per cent)
Baseline word model	50K	-	38.8
Vocabulary adaptation	-	56.6K	37.3
Predetermined vocabulary	56.6K	-	38.0

In vocabulary adaptation experiments, baseline vocabulary was modified with the word types in the extended lattice. Figures 5.9 – 5.10 indicate the relationship between OOV rates and WERs. As expected, the approach with the lower OOV rate achieves better recognition accuracy. Figure 5.10 leads us to conclude that for huge vocabulary sizes (the smallest adapted vocabulary size is 221 K), predetermined vocabularies perform better than adapted vocabularies in our task. This result is discouraging for the proposed technique. However, this can be the consequence of data sparseness caused by huge vocabulary sizes. The adapted vocabularies contain more rare words than the same size predetermined vocabularies. Therefore, data sparseness becomes a more crucial problem for vocabulary adaptation. For this reason, we performed the same experiment using a moderate size adapted vocabulary. To control the size of the adapted vocabulary, vocabulary adaptation was applied to the one best hypotheses instead of the lattice output of the 50 K system. The result of this experiment and the result for the same size predetermined vocabulary are given in Table 5.3. This finding demonstrates the effectiveness of vocabulary adaptation with a moderate size vocabulary. We also performed a cheating experiment to find the lower WER bound for vocabulary adaptation experiments. We used the 29 K fallback vocabulary, utilized in the lattice extension cheating experiment, as the adapted vocabulary. This vocabulary resulted in 30 per cent WER, which is our lower WER bound for the vocabulary adaptation experiments.

Cheating experiments for the proposed techniques are based on the ideal experimental scenarios. Since the set-ups for those experiments are different, the results are not directly comparable with each other. However, these experiments indicate that there is room for improvement for all of the techniques.

Since the main motivation in this work is to handle the high OOV rate problem, we analyzed the improvements obtained with the proposed techniques to understand if the gains are really coming from OOV handling. The analysis is performed using the error rate calculations given for OOV and IV words in Equations 4.1 and 4.2 respectively. This analysis reveals that (i) all the techniques reduce the WERs for OOV words (from 100 per cent to 67.3 per cent on average for word-based techniques, to 39.2 per cent for the morph baseline and to 34.6 per cent on average for the cheating experiments); (ii) the WER improvement for OOV words is more pronounced in vocabulary adaptation and predetermined vocabulary experiments; (iii) the WERs for IV words remain the same for lattice extension and predetermined vocabulary experiments, whereas WER for IV words increases for vocabulary adaptation; (iv) the lattice extension approach for morphs improve the WER for both the OOV and IV words compared to the morph baseline. This analysis shows that the gains are really coming from better OOV handling.

The OOV problem was best handled by using sub-word recognition units in this thesis. A shortcoming of sub-words is over-generation which may not be easily handled with morphological constraints in the case of statistical morphs. Therefore, lattice extension was modified for morphs to handle this shortcoming. Even though lattice extension for morphs brings back the OOV problem while generating the extended lattice from the lattice composed of word-like units, our best result was obtained with this approach. Lattice extension for morphs reduced the WER from 33.9 per cent to 32.3 per cent. When we analyzed the word-like units in the first-pass recognition output of the morph-based ASR system utilized in this Chapter, we found out that 159 out of 6759 word-like units did not occur in the fallback vocabulary and only 19 out of 159 were correct Turkish words. Invalid recognition outputs in morphs introduced 2.0 per cent WER to the baseline ASR system. The 1.6 per cent improvement obtained on morphs shows that lattice extension is mostly recovering the errors due to invalid words in the recognition output.

6. DISCRIMINATIVE LANGUAGE MODELING

Recent ASR systems utilize discriminative training methods on top of traditional generative models both for acoustic and language modeling. The advantage of discriminative parameter estimation to MLE is that discriminative training takes alternative (negative) examples into account as well as the correct (positive) examples. Therefore discriminative training estimates model parameters that discriminate well between different classes. In ASR framework, positive examples are the correct transcriptions and negative examples are the erroneous candidate transcriptions. Discriminative acoustic and language models utilize these examples to optimize an objective function that is directly related to the system performance. Discriminative acoustic model training utilize objective functions like Maximum Mutual Information (MMI) [73, 110] and Minimum Phone Error (MPE) [74] to estimate the acoustic model parameters that represent the discrimination between alternative classes. Discriminative language model (DLM) training aims to optimize the WER while learning the model parameters that discriminate the correct transcription of an utterance from the other candidate transcriptions.

Another advantage of DLM is that discriminative language modeling is a feature-based approach, like conditional random fields (CRFs) [111] and maximum entropy models [112], therefore, it allows for easy integration of relevant knowledge sources, such as morphology, syntax and semantics, into language modeling. As a result of improved parameter estimation with discriminative training and ease of incorporating overlapping features, discriminatively trained language models have been demonstrated to consistently outperform generative language modeling approaches [68, 69, 70, 71].

Discriminative language modeling is a complementary approach to the existing baseline generative language modeling. Baseline language models are utilized in the first-pass recognition to generate the ASR transcriptions (lattices or the N-best lists). DLMs map these transcriptions into real-valued d-dimensional feature vectors and then rerank the alternative hypotheses of each utterance with the discriminatively trained

feature parameters. The feature parameters are estimated from a separate training data which also consist of ASR transcriptions.

In this thesis, we apply the conventional approaches in generating ASR transcriptions for the training data, in mapping the transcriptions to d-dimensional feature vectors and in estimating the feature parameters. The novelty of this chapter is in the feature sets that try to extract the implicit information available at lexical and sub-lexical levels. The features proposed in this thesis take the morphological and syntactic structure of Turkish into account and derive useful information from data-driven sub-lexical units. The details of the feature sets will be explained after an introductory information on DLMs and will be followed by the experimental results.

6.1. Training Data Generation, Basic Features and Parameter Estimation

This section describes a general framework for discriminatively trained language models. We will follow the definitions and notations given in [68]. The main components of DLMs are as follows:

1. **Training Examples:** These are the input:output pairs (x_i, y_i) for $i = 1 \dots N$. Inputs, $x \in \mathcal{X}$, are the utterances and outputs, $y \in \mathcal{Y}$, are the corresponding reference transcriptions. Here $\mathcal{X}$ is the set of all possible inputs and $\mathcal{Y}$ is the set of all possible outputs.
2. $\boldsymbol{GEN(x)}$: This function enumerates a finite set of candidates for the inputs, x, where $GEN(x) \subseteq \mathcal{Y}$. $GEN(x)$ function can be the lattice or the N-best list output of the baseline ASR system for the utterance x.
3. $\boldsymbol{\Phi(x, y)}$: A d-dimensional real-valued feature vector ($\Phi(x, y) \in \Re^d$). The representation $\boldsymbol{\Phi}$ defines the mapping from the (x_i, y_i) pair to the feature vector $\Phi(x_i, y_i)$.
4. $\bar{\boldsymbol{\alpha}}$: A vector of discriminatively learned feature parameters ($\bar{\alpha} \in \Re^d$).

Like many other supervised learning approaches, DLM requires labeled input:output pairs as the training examples. Utterances with the reference transcriptions are uti-

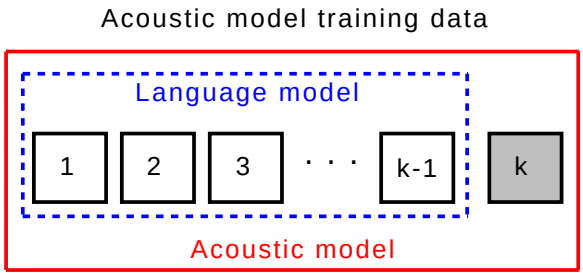


Figure 6.1. The training procedure for acoustic and language models for decoding the k'th fold of the acoustic model training data.

lized as the training examples, $(x_1, y_1) \ldots (x_N, y_N)$. These utterances are decoded with the baseline acoustic and language models in order to obtain the lattices or the N-best lists, in other words the candidate outputs for the $GEN(x)$ function. Since speech data with transcriptions are limited compared to the text data, it may not be possible to train the baseline acoustic and in-domain language models, and the DLM on separate corpora. This makes reducing the mismatch between training and test lattices a crucial point in DLM training. If the utterances of the training examples are decoded with the acoustic and language models trained on the same examples, then DLMs trained on this output lattice can not generalize to the test lattices. Note that the cheating experiment given in Table 3.3 reduces the WER from 23.4 per cent to 14.9 per cent when the decoding utterances are already seen in language modeling. Therefore, k-fold cross-validation can be applied on the training examples in baseline acoustic and language model training to alleviate the over-training of these models. However, acoustic model training is expensive and less prone to over-training than n-gram language model training. Therefore, cross-validation is only applied to the language model training. The lattices required for DLM training are generated by breaking the training examples into k folds, and recognizing the utterances in each fold using the baseline acoustic model (trained on all of the utterances) and an n-gram language model trained on the other k-1 folds. Figure 6.1 illustrates the training procedure for acoustic and language models for decoding the k'th fold, given in dark, of the acoustic model training data. The language model trained on the first k-1 folds and the acoustic model trained on the whole data are used to decode the utterances in the k'th fold.

Discriminative language modeling is a feature-based sequence modeling approach. Therefore, each (x_i, y_i) pair is mapped to a d-dimensional real-valued feature vector $\Phi(x_i, y_i)$. Each element of the feature vector, $\Phi_0(x_i, y_i) \ldots \Phi_{d-1}(x_i, y_i)$, corresponds to a different feature. Each candidate hypothesis of an utterance has a score from the baseline acoustic and language models. This score is used as the first element of the feature vector, $\Phi_0(x, y)$. This feature is defined as the "log-probability of y in the lattice produced by the baseline recognizer for utterance x". More formally $\Phi_0(x, y)$ is defined as

$$\Phi_0(x, y) = \beta \log P_l(y) + \log P_a(x|y) \tag{6.1}$$

where $\log P_l(y)$ and $\log P_a(x|y)$ are the baseline language model and acoustic model scores respectively and β is the language model weight.

The basic approach for the other DLM features is to use n-grams in defining features. The n-gram features are defined as the number of times a particular n-gram is seen in the candidate hypothesis. Different n-grams are represented as different features in the feature vector. Consider the Turkish phrase "`sunulacak bildiridekiler`" which means "*those in the paper that will be presented*". 2 unigrams and 1 bigram are obtained from this phrase and the n-grams are represented with the following features.

$\Phi_i(x, y)$ = number of times "`sunulacak`" is seen in y
$\Phi_j(x, y)$ = number of times "`bildiridekiler`" is seen in y
$\Phi_k(x, y)$ = number of times "`sunulacak bildiridekiler`" is seen in y

If there are d different features, the feature vector representation of the given phrase is $\Phi(x, y) = [\Phi_0(x, y) \;\; 0 \;\ldots\; 0 \; 1 \;\; 0 \;\ldots\; 0 \; 1 \; 1 \; 0 \;\ldots\; 0]^T$ where the first element of the feature vector, $\Phi_0(x, y)$, is the contribution of the baseline acoustic and language models and only the i'th, j'th and k'th terms of this feature vector have non-zero values. The feature values for these elements are one since these n-grams are seen only once in the given candidate hypothesis. Any type of information can be easily incorporated into language modeling via these features. In the next two sections, we will investigate

linguistically and statistically motivated feature types for Turkish.

Each DLM feature has an associated parameter, i.e., α_i for $\Phi_i(x, y)$. The best hypothesis under the $\bar{\alpha}$ model, y^*, maximizes the inner product of the feature and the parameter vectors, as given in Equation 6.2. The values of $\bar{\alpha}$ are learned in training and the best hypothesis under this model is searched for in decoding.

$$\begin{aligned} y^* &= \underset{y \in GEN(x)}{\text{argmax}} \ \langle \Phi(x, y), \ \bar{\alpha} \rangle \\ &= \underset{y \in GEN(x)}{\text{argmax}} \ (\ \alpha_0 \Phi_0(x, y) \ + \ \alpha_1 \Phi_1(x, y) \ + \ \cdots \ + \ \alpha_{d-1} \Phi_{d-1}(x, y) \ ) \end{aligned} \quad (6.2)$$

In this thesis a variant of the perceptron algorithm is utilized for parameter estimation (shown in Figure 6.2). This algorithm is the same with the one given in [68]. The main idea in this algorithm is to penalize features associated with the current 1-best hypothesis, and to reward features associated with the gold-standard hypothesis (reference or lowest-WER hypothesis). It has been found that the perceptron model trained with the reference transcription as the gold-standard hypothesis is much more sensitive to the value of the α_0 constant [68]. Therefore, we use the lowest-WER hypothesis (oracle) as the gold-standard hypothesis. In Figure 6.2, the training example (x_i, y_i) represents an utterance, x_i, and the corresponding gold-standard hypothesis, y_i. At the beginning of training, all the values of $\bar{\alpha}$ are set to 0. Then these parameters are updated during the passes (up to T-times) on the training examples. z represents one of the candidate hypothesis in the lattice or the N-best list for the utterance x_i, and z_i is the best scoring hypothesis under the current model. If the best scoring hypothesis is different than the gold-standard hypothesis, the values of the parameter vector are increased by the values in the feature vector of the gold-standard hypothesis and decreased by the values in the feature vector of the current best scoring hypothesis. The updated parameter vector, $\bar{\alpha}$, describes a hyperplane that tries to discriminate the gold-standard hypotheses from the other candidate hypotheses. Averaged parameters, $\bar{\alpha}_{AVG}$, are utilized in decoding held-out and test sets, since averaged parameters have been shown to outperform regular perceptron parameters in tagging tasks and also give much greater stability of the tagger [113]. In addition, there exist theoretical bounds

Inputs: Training examples (x_i, y_i) for $i = 1 \ldots N$

Initilization: Set $\bar{\alpha} = 0$

Algorithm:

For $t = 1 \ldots T$

For $i = 1 \ldots N$

Calculate $z_i = \underset{z \in GEN(x_i)}{\text{argmax}} \langle \Phi(x_i, z),\ \bar{\alpha} \rangle$

If$(z_i \neq y_i)$ then $\bar{\alpha} = \bar{\alpha} + \Phi(x_i, y_i) - \Phi(x_i, z_i)$

Output: Parameters $\bar{\alpha}$

Figure 6.2. A variant of the perceptron algorithm given in [68].

on the generalization error which are comparable to the bounds for Support Vector Machines (SVMs) [114]. If $\bar{\alpha}_t^i$ represents the feature parameters after the t'th pass on the i'th example, averaged parameters are defined as $\bar{\alpha}_{AVG} = \sum_{i,t} \bar{\alpha}_t^i / NT$.

6.2. Feature Sets for DLM with Words

This section describes the feature sets utilized in Turkish DLMs with words. The features are extracted from the N-best list output of the training utterances decoded by the word-based ASR system. In this thesis we investigate word n-grams, grammatical and statistical sub-lexical units and syntactic information to define the feature sets.

6.2.1. Word n-gram Features

Word n-grams are the basic features utilized in our experiments. The details of this feature set was explained in Section 6.1.

6.2.2. Sub-lexical Features

Using sub-lexical units instead of words in generative language modeling is a common approach in baseline ASR systems for agglutinative languages. As was mentioned in Chapter 4, grammatical units obtained with the morphological analysis and statistical units obtained with the Morfessor Algorithm can be used as the sub-lexical

approaches in generative language models. In DLMs, these sub-lexical units are utilized in defining the features to rerank the N-best word hypotheses. In this section, the features are determined on the sub-lexical sequences after converting word hypothesis sentences into sub-lexical sequences by morphological or statistical decompositions. Section 6.3 will define the features on sub-lexical hypothesis sentences that are directly obtained from the ASR system.

6.2.2.1. Grammatical sub-lexical features. Morphology is an important information source for feature-based language models, especially for morphologically rich languages [71, 115]. Turkish has a rich morphological structure that introduces challenges for ASR systems (See Section 2.2.2). In this section, we aim to turn this challenging structure into a useful information source when reranking N-best word hypotheses with DLMs. Therefore, this section focuses on extracting information from morphological decompositions and utilizing this information as DLM features.

The grammatical sub-lexical features will be explained using the example phrase and the corresponding morphological analysis given in Figure 6.3. The morphological analysis was obtained with Oflazer's morphological analyzer [92]. The ambiguity in the words with multiple morphological analysis was resolved using the perceptron-based morphological disambiguation tool for Turkish [116]. The details of the morphological parser and the disambiguation tool were explained in Chapter 3.

In Figure 6.3, morphological information is grouped in order to obtain useful groupings for DLM features. Endings are the groupings of the surface form morphemes and inflectional groups are the groupings of the consecutive morphological tags. In the morphological analysis, the tag sequences are separated by the ^DB symbol which denotes the derivation boundaries. The sub-lexical units separated by derivation boundaries are called the Inflectional Groups (IGs). As shown in Figure 6.3, a Turkish word can be represented as a root and a sequence of IGs or a root and an ending[16]. Note that the morphological tags following the root of the word until the derivation

[16] Some Turkish words can be simple roots without any endings.

sunulacak bildiridekiler *(those in the paper that will be presented)*

Morphological Analysis:

sunulacak:
  sun+Verb+^DB +Verb+Pass+Pos+^DB +Adj+FutPart+Pnon

bildiridekiler:
  bildiri+Noun+A3sg+Pnon+Loc+^DB +Adj+Rel+^DB +Noun+Zero+A3pl+Pnon+Nom

Roots and Endings:

r_1 = root (sunulacak) = sun *(present)*

e_1 = ending (sunulacak) = -ulacak

r_2 = root (bildiridekiler) = bildiri *(paper)*

e_2 = ending (bildiridekiler) = -dekiler

Roots and Inflectional Groups (IGs):

r_1 = root (sunulacak) = sun *(present)*

$IG_{1,1}$ = IG1 (sunulacak) = +Verb

$IG_{1,2}$ = IG2 (sunulacak) = +Verb+Pass+Pos

$IG_{1,3}$ = IG3 (sunulacak) = +Adj+FutPart+Pnon

r_2 = root (bildiridekiler) = bildiri *(paper)*

$IG_{2,1}$ = IG1 (bildiridekiler) = +Noun+A3sg+Pnon+Loc

$IG_{2,2}$ = IG2 (bildiridekiler) = +Adj+Rel

$IG_{2,3}$ = IG3 (bildiridekiler) = +Noun+Zero+A3pl+Pnon+Nom

Figure 6.3. Example Turkish phrase with morphological analysis. Endings and IGs are the groupings of the morphological information.

boundary is the first IG of this word. If there is no derivation boundary in a word, then this word is decomposed into a root and a subsequent IG. For instance the i'th word, (w_i), in a phrase is represented as

$$r_i + IG_{i,1} + \text{DB} + ... + IG_{i,j} + \text{DB} + ... + IG_{i,n_i}$$

where r_i is the root of w_i, n_i is the number of inflectional groups in w_i and IG_{i,n_i} is the last inflectional group of w_i. $IG_{i,j}$ represents the j'th inflectional group of w_i where $1 \leq j \leq n_i$. The words in Figure 6.3 are analyzed as a root and three IGs.

The grammatical morphemes, in surface or lexical forms, are not provided in

the morphological analyser output and there is not a one-to-one mapping between the morphological tags and the morphemes. For instance the morphological tag sequence `+Noun+A3sg+Pnon+Loc` correspond to the lexical morpheme `+dA`[17], however, the morphological tag sequence `Noun+A3sg+Pnon+Nom` represents a nominal noun and does not have any matching morpheme. Therefore, we only investigate the information obtained directly from the analyser in defining the morphological features. In this thesis, we have used root, stem+ending and IG-based n-grams as the grammatical sub-lexical features. We do not use each morphological tag as a separate component in feature extraction since a grouping, an ending or an IG, bears more information than a single morphological tag. In order to obtain the features, all the words in the hypothesis sentences are morphologically analyzed and disambiguated. The words that can not be analyzed with the parser are left as unparsed and represented as nominal nouns.

In **root n-gram features**, first the words in the hypothesis sentences are represented only with their roots using the morphological decompositions. Then the n-gram features are generated in the same way with words as if roots are words. The root unigram and bigram feature templates are listed below with examples from Figure 6.3. r_i represents the root of the i'th word.

1. Unigram: ($\mathtt{r_i}$)

   Example for ($\mathtt{r_2}$):

   $\Phi_k(x, y)$ = number of times "`bildiri`" is seen in y
2. Bigram: ($\mathtt{r_{i-1}\ r_i}$)

   Example for ($\mathtt{r_1\ r_2}$):

   $\Phi_k(x, y)$ = number of times "`sun bildiri`" is seen in y

In **stem+ending n-gram features**, the stem is extracted from the morphological decomposition and the remaining part of the word is taken as the ending. If there is no ending in the word, a special symbol is inserted to represent the empty ending. Hypothesis sentences are converted to stem and ending sequences and the n-gram features are generated in the same way with words as if stems and endings are

[17] 'A' is the lexical symbol realized as /a/ or /e/ in surface form.

words. The stem+ending unigram and bigram feature templates are listed below with examples from Figure 6.3. r_i and e_i represent the root and the ending of the i'th word respectively.

1. Unigrams: ($\texttt{r}_\texttt{i}$) and ($\texttt{e}_\texttt{i}$)

   Example for ($\texttt{e}_\texttt{2}$):

   $\Phi_k(x, y)$ = number of times "`-dekiler`" is seen in y
2. Bigrams: ($\texttt{r}_{\texttt{i}-1}$ $\texttt{e}_{\texttt{i}-1}$) and ($\texttt{e}_{\texttt{i}-1}$ $\texttt{r}_\texttt{i}$)

   Example for ($\texttt{e}_\texttt{1}$ $\texttt{r}_\texttt{2}$):

   $\Phi_k(x, y)$ = number of times "`-ulacak bildiri`" is seen in y

IG-based features are commonly used to handle the challenges introduced by the agglutinative nature in dependency parsing and morphological disambiguation tasks of Turkish [93, 96, 116]. In previous studies it has been shown that the dependency relations between words are determined by the relations between the last IG of a word and any inflectional groups of the word on the right [96]. Moreover, IG-based features in a discriminative framework give the best accuracy for the Turkish morphological disambiguation and PoS tagging tasks [116]. Therefore, IGs are also utilized in defining the morphological features in this research. The words in the hypothesis sentences are represented as a sequence of roots and IGs and the IG-based n-gram features are extracted from these sequences. Note that IG-based features also contain the roots of the words in the feature definitions. The IG-based unigram and bigram feature templates are listed below with examples from Figure 6.3. r_i and $IG_{i,j}$ represent the root and the j'th IG of the i'th word respectively.

1. Unigrams: ($\texttt{r}_\texttt{i}$) and ($\texttt{IG}_{\texttt{i,j}}$)

   Example for ($\texttt{IG}_{\texttt{2,3}}$):

   $\Phi_k(x, y)$ = number of times "`+Noun+A3sg+Pnon+Loc`" is seen in y
2. Bigrams: ($\texttt{r}_{\texttt{i}-1}$ $\texttt{IG}_{\texttt{i}-1,1}$), ($\texttt{IG}_{\texttt{i}-1,\texttt{j}-1}$ $\texttt{IG}_{\texttt{i}-1,\texttt{j}}$) and ($\texttt{IG}_{\texttt{i}-1,\texttt{n}_{\texttt{i}-1}}$ $\texttt{r}_\texttt{i}$)

   Example for ($\texttt{IG}_{\texttt{1,3}}$ $\texttt{r}_\texttt{2}$):

   $\Phi_k(x, y)$ = number of times "`+Adj+FutPart+Pnon bildiri`" is seen in y
3. Bigrams in consecutive words: ($\texttt{r}_{\texttt{i}-1}$ $\texttt{r}_\texttt{i}$), ($\texttt{r}_{\texttt{i}-1}$ $\texttt{IG}_{\texttt{i,j}}$), ($\texttt{IG}_{\texttt{i}-1,\texttt{j}}$ $\texttt{r}_\texttt{i}$) and

($\texttt{IG}_{\texttt{i}-1,\texttt{j}}$ $\texttt{IG}_{\texttt{i,j}}$)

Example for ($\texttt{r}_1$ $\texttt{IG}_{2,3}$):

$\Phi_k(x, y)$ = number of times "`sun +Noun+A3sg+Pnon+Loc`" is seen in y

4. Bigrams from the last inflectional group of the previous word to all the inflectional groups of the current word: ($\texttt{IG}_{\texttt{i}-1,\texttt{n}_{\texttt{i}-1}}$ $\texttt{IG}_{\texttt{i,j}}$)

   Example for ($\texttt{IG}_{1,3}$ $\texttt{IG}_{2,2}$):

   $\Phi_k(x, y)$ = number of times "`+Adj+FutPart+Pnon +Adj+Rel`" is seen in y

IG bigrams take the intra-word and cross-word IG relations and IG bigrams in consecutive words take all the inter-word IG relations into account. The last feature set is the subset of the "bigrams in consecutive words". The main motivation in this feature set is to decrease the number of features using the knowledge that ($\texttt{IG}_{\texttt{i}-1,\texttt{n}_{\texttt{i}-1}}$ $\texttt{IG}_{\texttt{i,j}}$) pairs are more likely to contain the potential dependency relations between words.

6.2.2.2. Statistical sub-lexical features. Statistical morphs obtained with the Morfessor algorithm are used in defining the statistical sub-lexical features. The Morfessor algorithm is a data-driven approach. As a result, the algorithm does not require any linguistic information in splitting words into morph sequences. This is the main advantage of the Morfessor algorithm over a morphological parser. However, as a consequence of ignoring the linguistic knowledge in generating the decompositions, morphs do not convey any explicit morphological information like grammatical morphemes. We investigate morphs as DLM features since these units are much easier to obtain than the grammatical units and we want to explore their effectiveness in DLMs.

In order to obtain the morph features from the word hypotheses, the words in the candidate hypotheses are replaced with their corresponding morph segmentations. The details of obtaining the morph segmentations from the word vocabulary was explained in Chapter 4. An example hypothesis sentence with the morph segmentations is given in Figure 6.4.

The **morph n-gram features** are extracted from the morph sequences in the

Word hypothesis:

`sunulacak bildiridekiler` *(those in the paper that will be presented)*

Word hypothesis segmented into statistical morphs:

`sunul -acak bildir -ide -kiler`

Figure 6.4. A word hypothesis sentence segmented into statistical morphs.

same way with words as if morphs are words. The morph unigram and bigram feature templates are listed below with examples from Figure 6.4. m_i represents the i'th morph in the sentence.

1. Unigram: ($\mathtt{m_i}$)

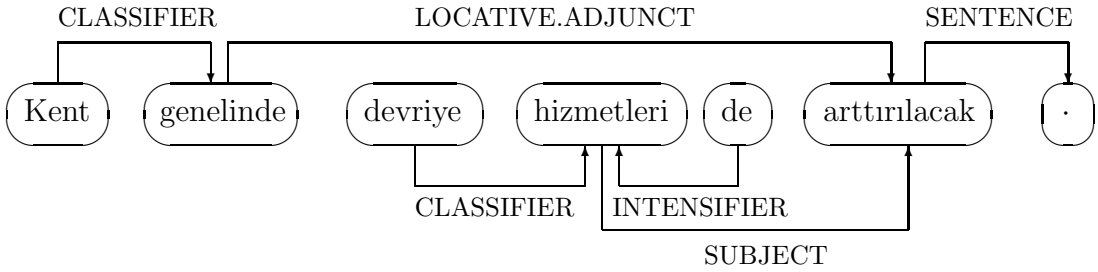
   $\Phi_k(x, y)$ = number of times "`-acak`" is seen in y
2. Bigrams: ($\mathtt{m_{i-1}}$ $\mathtt{m_i}$)

   $\Phi_k(x, y)$ = number of times "`-acak bildir`" is seen in y

6.2.3. Syntactic Features

Syntax is an important information source for language modeling due to its role in sentence formation. Syntactic information has been incorporated into conventional generative language models using left-to-right parsers to capture long distance dependencies in addition to $n-1$ previous words [59, 60]. Syntactic information have also been utilized in feature-based reranking approaches [70, 67, 64]. The success of these approaches lead us to investigate syntactic features for Turkish DLMs. Additionally, Section 4.3.2 reported the error analysis of the best scoring baseline system, the morph-based baseline system. In this analysis, we found out that majority, 99 per cent, of the errors labelled as correctable were coming from syntactic errors and these errors were recognized correctly in the gold-standard hypothesis. Therefore we aim to reach the syntactically correct hypothesis while reranking the candidate hypotheses with the syntactic DLM features.

For the syntactic DLM features, we explore similar feature definitions with [70].



WORD:	Kent	genelinde	devriye	hizmetleri	de	arttırılacak	·
GLOSS:	The city	throughout	patrol	services	also	will be increased	·
PoS:	[Noun]	[Noun]	[Noun]	[Noun]	[Conj]	[Verb]	[Punc]

Figure 6.5. Example dependency analysis for syntactic features.

We used PoS tag n-grams and head-to-head (H2H) dependency relations between lexical items or their PoS tags as the syntactic features. PoS tag features are utilized in an effort to obtain class-based generalizations that may capture well-formedness tendencies. H2H dependency relations are utilized since presence of a word or morpheme can depend on the presence of another word or morpheme in the same sentence and this information is represented in the dependency relations.

The syntactic features will be explained with the dependency analysis given in Figure 6.5 for the Turkish sentence, which means "Patrol services will also be increased throughout the city". The incoming and outgoing arrows in the figure show the dependency relations between the head and the dependent words with the type of the dependency. The words with English glosses, PoS tags associated with the words are also given in the example. Eryiğit et al.'s dependency parser [96] is used for the dependency analysis. Figure 6.5 was also given in Chapter 3 to explain Eryiğit et al.'s dependency parser and reproduced here for the demonstration of the syntactic DLM features.

The syntactic feature templates are listed below with examples from Figure 6.5. $\mathtt{t_i}$ represents the PoS tag of the i'th word and $DR(w_i, w_k)$ represents the type of the dependency relation between the i'th and the k'th words.

1. Unigrams: $(\mathtt{t_i})$

Example for the word '`Kent`':

$\Phi_k(x, y)$ = number of times "`[Noun]`" is seen in y

2. Bigrams: ($\texttt{t}_{\texttt{i}-1}$ $\texttt{t}_{\texttt{i}}$)

   Example for the words '`hizmetleri de`':

   $\Phi_k(x, y)$ = number of times "`[Noun][Conj]`" is seen in y

3. Head-to-Head (H2H) dependencies:

   Examples for the words '`Kent genelinde`':

   - dependencies between lexical items: $(DR(w_i, w_k)\ w_i\ w_k)$

     $\Phi_k(x, y)$ = number of times "`CLASSIFIER Kent genelinde`" is seen in y

   - dependencies between a single lexical item and the PoS of another item: $(DR(w_i, w_k)\ w_i\ t_k)$ and $(DR(w_i, w_k)\ t_i\ w_k)$

     $\Phi_k(x, y)$ = number of times "`CLASSIFIER Kent [Noun]`" is seen in y

     $\Phi_l(x, y)$ = number of times "`CLASSIFIER [Noun] genelinde`" is seen in y

   - dependencies between PoS tags of lexical items: $(DR(w_i, w_k)\ t_i\ t_k)$

     $\Phi_k(x, y)$ = number of times "`CLASSIFIER [Noun] [Noun]`" is seen in y

To obtain the syntactic features from the training examples, the dependency analyses of hypothesis sentences are derived. Then PoS tag and H2H features are extracted from these dependency analyses. Here, it is important to note that hypothesis sentences contain recognition errors and the parser generates the best possible dependency relations even for incorrect hypotheses.

6.3. Feature Sets for DLM with Sub-lexical Units

Section 6.2.2 explored sub-lexical features to rerank the N-best word hypotheses. In this section, sub-lexical units are used as language modeling items in the baseline ASR system to generate the hypothesis sentences composed of sub-lexical units. DLM features are extracted from these sub-lexical sequences and they are utilized to rerank the N-best sub-lexical hypotheses. Only the statistical morph-based system that yields the best recognition accuracy in Chapter 4 (non-initial morphs marked with '-') is used as the sub-lexical approach in DLM research.

The main motivation in utilizing DLMs with sub-lexical units is to investigate the interaction between language modeling units and discriminative training. We thought that there can be an interaction between these two components since generating the negative examples for discriminative training makes use of the baseline language model. Therefore, using basic word features on word hypotheses and using basic morph features on morph hypotheses in discriminative language modeling would let us to investigate this potential interaction. We also explore morpho-syntactic features in addition to the basic morph n-gram features. Morpho-syntactic features are motivated by the syntactic features explained in Section 6.2.3 and extracted from the morph sequences with data-driven approaches. The details of the proposed feature sets will be explained in the following sections.

6.3.1. Sub-lexical n-grams

Morph n-grams are the basic sub-lexical features. Morph n-gram features are defined in the same way with the sub-lexical features given in Section 6.2.2.2. However, in this section the morph unigram and bigram features are directly obtained from the morph hypotheses instead of segmenting the word hypotheses into morph sequences. Figure 6.6 shows an example morph hypothesis sentence and the same sentence where the morph sequences are converted into word-like units. Word-like units are obtained by concatenating non-initial morphs to the previous morphs. The morph hypothesis sentence in the figure produces a grammatically correct Turkish phrase which means "*those in the paper that will be presented*". However, this is not always the case, since morph sequences can also yield ungrammatical or non-word items.

Morph hypothesis:

`sunul -acak bildir -ide -kiler`

Morph sequences converted into word-like units:

`sunulacak` `bildiridekiler` *(those in the paper that will be presented)*

Figure 6.6. A morph hypothesis sentence converted into word-like units using the non-initial morph markers.

In addition to the classical morph n-gram features, we propose using only the **word internal morph bigrams** as DLM features to deal with the non-word morph sequences in the morph hypotheses. This feature set is the subset of the morph bigram features where cross-word bigrams, e.g., "`-acak bildir`", are excluded.

In word or root n-gram features, the bigrams cover the relations between consecutive words. However, the morph bigrams only cover the intra-word and cross-word relations since each word is segmented on average 1.4 morphs. Therefore n-gram orders higher than words were utilized in generative morph language models. 3-gram word and 4-gram morph language models yielded the best accuracy in the baseline ASR systems. We can also utilize higher order morph n-gram features in DLMs to capture the information in consecutive words. However, the drawback of using higher order n-gram features is that the amount of DLM training examples is very small compared to the text corpora, as a result, higher order n-gram features can suffer from the sparsity problem due to the small number of observations per parameter. Therefore we propose **first-morph bigram features**, where only the bigrams between the first morphs of consecutive words are extracted, in an effort to cover the word bigram context with morph bigrams. In this feature set, the main idea is the same with the first-morph-based similarity criterion in Section 5.1.3.2. The first morph of each word is considered as the most semantic information bearing part of that word like roots. So, we make an analogy between root bigram features and first-morph bigram features.

The morph n-gram feature templates are listed below with examples from Figure 6.6. $\mathtt{m_{i,j}}$ represents the j'th morph of the i'th word where $1 < j < n_i$. $\mathtt{m_{i,n_i}}$ represents the last morph of the i'th word.

1. Unigrams: ($\mathtt{m_{i,j}}$)

   $\Phi_k(x, y)$ = number of times "`sunul`" is seen in y
2. Bigrams: ($\mathtt{m_{i,j-1}}$ $\mathtt{m_{i,j}}$) and ($\mathtt{m_{i-1,n_i}}$ $\mathtt{m_{i,1}}$)

   $\Phi_k(x, y)$ = number of times "`sunul -acak`" is seen in y

   $\Phi_l(x, y)$ = number of times "`-acak bildir`" is seen in y
3. Word internal bigrams: ($\mathtt{m_{i,j-1}}$ $\mathtt{m_{i,j}}$)

$\Phi_k(x, y)$ = number of times "`bildir -ide`" is seen in y

4. First morph bigrams: ($\mathtt{m}_{\mathtt{i}-1,1}$ $\mathtt{m}_{\mathtt{i},1}$)

   $\Phi_k(x, y)$ = number of times "`sunul bildir`" is seen in y

6.3.2. Morpho-Syntactic Features

The advantage of the statistical morphs compared to their grammatical counterparts is that they do not require linguistic knowledge for segmenting words into sub-lexical units. As a result morphs do not convey explicit linguistic information like grammatical morphemes and obtaining linguistic information from morph sequences is not obvious. One way of information extraction from morphs is to convert them into word-like units and to apply the same procedure with words. However, this indirect approach tends to fail when concatenation of morph sequences does not generate grammatically correct words. In addition, this approach contradicts with the main idea of statistical morphs – obtaining sub-lexical units without any linguistic tools. In this section we focus on exploring representative features of implicit morpho-syntactic information in morph sequences. We explore morph-based features similar to PoS tag and H2H dependency features using data driven approaches. The details of these approaches will be explained in the following sections.

6.3.2.1. Clustering of Sub-lexical Units. The feature set proposed in this section aims to obtain syntactic information, similar to PoS tags, directly from the morphs in the hypothesis sentences. PoS tags determine the syntactic categories of words, in other words, classify words according to their function in the context. Linguistic tools are utilized to obtain the PoS tags of words in a sentence. If morph sequences produce grammatically correct words, the PoS tags associated with the word-like units can be obtained in the same way. However, we can not assign a PoS tag to an individual morph using linguistic tools. In order to obtain categories for morphs, like PoS tags of words, morphs in the training corpora are automatically classified with clustering algorithms. The category associated with a particular morph is considered as the tag of that morph and utilized in defining the morpho-syntactic features.

We apply two hierarchical clustering approaches on morphs to obtain meaningful categories. The first one is Brown et al.'s algorithm [58] which aims to cluster words into semantically-based or syntactically-based groupings by maximizing the average mutual information of adjacent classes. Brown et al.'s algorithm is proposed for class-based n-gram language models and the optimization criterion in clustering is directly related to the n-gram language model quality. Utilizing n-gram features in DLMs makes this clustering an attractive approach for our research. The second approach utilizes minimum edit distance (MED) as the similarity function in bottom-up clustering. The motivation in this algorithm is to capture the syntactic similarity of morphs using their graphemic similarities, since a non-initial morph can cover a grammatical morpheme, a group of grammatical morphemes or pieces of grammatical morphemes. In our application we modify MED to softly penalize the variations in the lexical and surface forms of morphemes.

Class-based language modeling aims to handle the data sparseness problem in parameter estimation by group words that have similar syntactic or semantic properties. The word n-gram probabilities in class-based language models are calculated as

$$P(w_i|w_{i-n+1}\dots w_{i-1}) = P(w_i|c_i)P(c_i|c_{i-n+1}\dots c_{i-1}) \tag{6.3}$$

where c_i is the class of the i'th word w_i. The clustering algorithm yields the assignment of the words into clusters. This assignment is defined as a function $\pi(\cdot)$ where $c_i = \pi(w_i)$. Now we will explain Brown et al.'s clustering algorithm using the definitions and notations given in [58].

$L(\pi)$ is the log likelihood of the training data as a function of π and defined as

$$L(\pi) \quad = \quad \frac{1}{(T-1)} \log P(s_2^T|s_1) \tag{6.4}$$

where s_2^T represents a sequence of strings from s_2 to s_T, i.e., $s_2\ s_3 \dots s_{T-1}\ s_T$. Each string ($s_i$) corresponds to a word when clustering words and corresponds to a morph when clustering morphs. Using the class-based bigram approximations in the chain

rule, $P(s_i|s_1 \ldots s_{i-1}) \approx P(s_i|s_{i-1}) = P(s_i|c_i)P(c_i|c_{i-1})$, Equation 6.4 turns into

$$L(\pi) = \frac{1}{(T-1)} \sum_{i=2}^{i=T} \log P(s_i|c_i)P(c_i|c_{i-1}) \tag{6.5}$$

In the limit, this equation reduces to[18]

$$L(\pi) = \sum_{c_1 c_2} P(c_1 c_2) \log \frac{P(c_1 c_2)}{P(c_1)P(c_2)} + \sum_{s} P(s) \log P(s) \tag{6.6}$$

$$= I(c_1, c_2) - H(s) \tag{6.7}$$

where $I(c_1, c_2)$ is the average mutual information between adjacent classes and $H(s)$ is the entropy of the unigram string distribution in the training data. In Equation 6.6, $\sum_{c_1 c_2}$ represents the summation over all possible cluster bigrams and $\sum_s$ represents the summation over all possible unigrams in the training data. $L(\pi)$ depends on π only through $I(c_1, c_2)$ since the mapping π determines the partitioning of the strings in the training data into clusters. Here the objective is to maximize the likelihood $L(\pi)$ and this can be achieved by maximizing the average mutual information of adjacent classes. However, there is no practical way of obtaining the clusters that maximize this mutual information. Therefore a greedy algorithm is utilized. Initially all word types are assigned to distinct clusters. Then cluster pairs are merged if the loss in average mutual information is least. After a few successive merges, individual words are reassigned to other clusters to find another partitioning that increases the average mutual information of adjacent classes. The algorithm stops if there is no potential reassignment of a word leading to a patition yielding higher average mutual information. The algorithm does not consider the simultaneous reassignments of two or more words in rearranging the partitions to achieve higher average mutual information. This makes the algorithm feasible, however, it can result in a sub-optimal partitioning of the training data.

In this research, we apply this clustering approach for assigning morphs into a

[18] See Appendix A for the derivation of Equation 6.7 from Equation 6.4.

predefined number of clusters. Brown et al.'s clustering algorithm has already been implemented in SRILM toolkit. Therefore, we utilize SRILM toolkit for the clustering algorithm. We perform trials with 50, 100 and 200 clusters. Appendix B gives the distribution of initial and non-initial morphs in each class with some examples for the trial with 50 classes. We can not explicitly state if the morphs in the classes are syntactically or semantically related by looking at only a few samples. However, our observation is that there can be a syntactic relationship between non-initial morphs occurring in the same class for the classes where the number of non-initial morphs is much higher than the number of initial morphs. The graphemic similarity of the non-initial morphs in these classes indicate that non-initial morphs can have common morphemes, since some of the non-initial morphs coincide with the grammatical morphemes or their groupings. We have also observed that the initial morphs having similar meanings are grouped together. For instance, we search for the days of the week, that are left as initial morphs in the morph segmentations (six out of seven are left as initial morphs), and it is promising to see that five of them are grouped in the same class. Only the day "pazar" (*sunday*) is misclassified most probably due to being the synonym of a common word "pazar" (*market*). Additionally, the 29. class contains the digits and the 21. class contains the name of the months as the initial morphs in Appendix B.

Our second clustering approach utilizes MED similarity in bottom-up clustering. This clustering is only meaningful for non-initial morphs since graphemic similarity of initial morphs does not reveal any linguistic information. Therefore, we only cluster the non-initial morphs and all the initial morphs are assigned to the same cluster. In this clustering algorithm, initially all the non-initial morphs are assigned to distinct clusters and the cluster pair that has the highest similarity is merged at every iteration of the algorithm. The similarity of two different clusters are defined with the similarity between the members of these clusters. In our research we use the complete-link similarity function where the similarity of two clusters are measured with the similarity of the two least similar members in the clusters. In contrast, single-link similarity function defines the similarity of two clusters with the similarity of the two most similar members. We prefer complete-lnik to single-link similarity function to obtain tight clusters instead of "straggly" clusters [117].

Let's define $d(c_i, c_j)$ as the distance between two clusters c_i and c_j. $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ is the set of the clusters and $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ is the set of input strings that will be assigned to the clusters in $\mathcal{C}$. In our case, s_i is one of the non-initial morphs in the vocabulary and $N = 27K$. In bottom-up hierarchical clustering, initially k is set to N and each non-initial morph is assigned to a distinct cluster. After $N - 1$ iterations all the strings are grouped into a single cluster ($k = 1$). If $d(s, s')$ represents the distance between the strings s and s', then $d(c_i, c_j)$ is as follows:

$$
\begin{aligned}
(s_i, s_j) &= \underset{(s,s') \in c_i \times c_j}{\operatorname{argmax}} d(s, s') && (6.8)\\
d(c_i, c_j) &= d(s_i, s_j) && (6.9)
\end{aligned}
$$

where $(s, s') \in c_i \times c_j$ represents all the string pairs generated with the strings that are already clustered in the classes c_i and c_j. We choose the string pair, (s_i, s_j), yielding the highest distance for defining the similarity of the clusters because this string pair gives the most dissimilar members of the clusters for the complete-link similarity function. Equation 6.10 shows the merging step of the algorithm. First the cluster pair yielding the lowest distance, highest similarity, is chosen among the pairs in $\mathcal{C}$ x $\mathcal{C}$. Then the clusters in the pair are merged into a single cluster and the cluster set $\mathcal{C}$ is updated.

$$
\begin{aligned}
(c_i, c_j) &= \underset{(c,c') \in \mathcal{C} \times \mathcal{C}}{\operatorname{argmin}} d(c, c')\\
c_{new} &= c_i \cup c_j\\
\mathcal{C} &= \mathcal{C} \setminus \{c_i, c_j\} \cup c_{new} && (6.10)
\end{aligned}
$$

The distance function, $d(s, s')$, in this algorithm is the minimum edit distance (MED) between the input strings. The MED equation is given in Equation 5.3. To make the distance function independent of the string lengths, we normalize the MED of the input strings with their string lengths, i.e., $\frac{d(s,s')}{|s|} + \frac{d(s,s')}{|s'|}$ where $|s|$ is the length of the string s. If x and y represent strings with arbitrary lengths and x^t denotes the letter at the t'th position, the costs of the edit operations for insertion, $c(\epsilon, x^t)$,

deletion, $c(x^t, \epsilon)$, and substitution, $c(x^t, y^v)$, are defined as follows in Section 5.1.3.3.

$$c(x^t, \epsilon) = c(\epsilon, y^v) = 1 \text{ and } c(x^t, y^v) = \begin{cases} 1 \text{ if } x_t \neq y_v, \\ 0 \text{ if } x_t = y_v. \end{cases} \tag{6.11}$$

As was mentioned in Chapter 2.2.1, the same lexical form morpheme can correspond to different surface form morphemes due to vowel harmony, consonant harmony, and consonant deletion rules of Turkish. Even though there is not an exact match between the statistical morphs and the grammatical morphemes of the same word, a statistical morph can contain a grammatical morpheme or a group of grammatical morphemes. Therefore, considering the phonological variations in surface and lexical form morphemes in clustering will help grouping morphs with similar syntactic functions into the same cluster. To do that we rearrange the substitution, deletion and insertion costs in the MED calculations to softly penalize the potential phonological variations in the surface and lexical forms of the same suffix. The substitutions between the vowel pairs "a-e", "ı-i", and the consonant pair "t-d" are penalized with a cost of 0.5 to take the vowel harmony and the consonant harmony into account. The deletions or insertions between the pairs "ϵ-y", "ϵ-n", "ϵ-s" are also penalized with a cost of 0.5 to take the consonant deletions into account during suffixation. Note that only the phonological variations in the suffixes are taken into account since we are trying to cluster only the non-initial morphs. Example sub-trees are given in Figure 6.7.

After clustering morphs with one of the hierarchical clustering algorithms, the cluster of each morph is taken as the tag of that morph. The n-gram DLM features are extracted on the morph tag sequences. The feature templates are listed below. $\texttt{c}_\texttt{i}$ represents the cluster of the i'th morph, $\texttt{m}_\texttt{i}$, and π represents the mapping from morphs to clusters, i.e., $c_i = \pi(m_i)$.

1. Unigram: $(\texttt{c}_\texttt{i})$

   $\Phi_k(x, y)$ = number of times "$\pi($`-acak`$)$" is seen in y
2. Bigrams: $(\texttt{c}_{\texttt{i}-1}\ \texttt{c}_\texttt{i})$

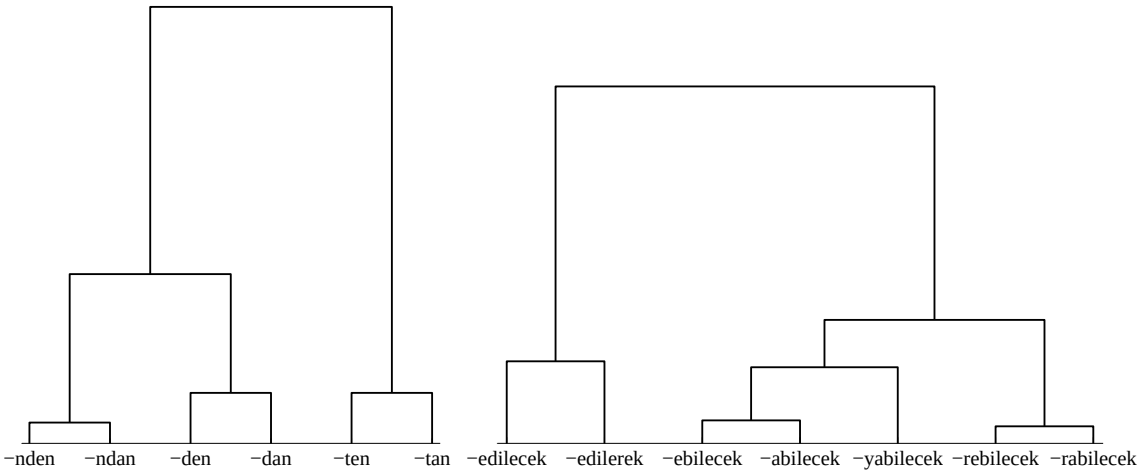


Figure 6.7. Example sub-trees

$$\Phi_k(x, y) = \text{number of times “}\pi(\texttt{-acak})\ \pi(\texttt{bildir})\text{” is seen in } y$$

6.3.2.2. Long Distance Triggers. In addition to morph clusters, we also propose long distance triggers as morpho-syntactic features for DLMs with sub-lexical units. These features are motivated by the H2H dependency features in words. Considering initial morphs as stems and non-initial morphs as suffixes, we assume that the existence of a morph can trigger another morph in the same sentence. The morphs in trigger pairs are believed to co-occur for a syntactic function, like the syntactic dependencies of words, and these pairs are utilized to define the long distance morph trigger features

Long distance morph trigger features are similar to the trigger features proposed in [63, 72]. In [63], trigger pairs are utilized as features to adapt the expectations of the maximum entropy language models to the topic of discourse. In [72], trigger features are utilized in DLMs to capture the re-occurrence of the words using the conversation context. Both of the approaches use the document history to find the trigger features. In that respect using trigger features in language modeling resembles the cache language models [118] proposed to capture the dynamic nature of the n-grams.

In our research, we utilize trigger features in DLMs with sub-lexical units. Ad-

Morph hypothesis:

`sunul -acak bildir -ide -kiler`

Candidate trigger pairs:

`sunul -acak` `sunul bildir` `sunul -ide` `sunul -kiler`

`-acak bildir` `-acak -ide` `-acak -kiler`

`bildir -ide` `bildiri -kiler`

`-ide -kiler`

Figure 6.8. A morph hypothesis sentence and the candidate trigger pairs extracted from this hypothesis.

ditionally, we only consider sentence level trigger pairs to capture the syntactic-level dependencies instead of discourse-level information. These two details make our approach different than the previous applications of the trigger features in feature-based language models. In a word hypothesis, the dependency relations can be good syntactic trigger pairs. Since these relations are not directly extracted from morph sequences, we extract all the morph pairs between the morphs of any two words in a sentence as the candidate morph triggers. The candidate morph trigger pairs are extracted from the hypothesis sentences (1-best and oracle) to obtain also the negative examples for DLMs. See Appendix C for the examples of morph trigger pairs.

An example morph hypothesis sentence with the candidate trigger pairs are given in Figure 6.8. Among the possible candidates, we try to select only the pairs where morphs are occurring together for a special function. This is formulated with hypothesis testing where null hypothesis (H_0) represents the independence and the alternative hypothesis (H_1) represents the dependence assumptions of morphs in the pairs [117]. The pairs with higher likelihood ratios ($\log \frac{L(H_1)}{L(H_0)}$) are assumed to be the morph triggers and utilized as features. The feature template is listed below. In the example morph trigger feature given below, we consider that among the candidate pairs in Figure 6.8, the pair "`sunul bildir`" is one of the morph triggers. $(m_i \rightarrow m_j)$ represents the trigger pair and $i < j$ for all the pairs.

1. Bigrams: $(m_i \rightarrow m_j)$

   $\Phi_k(x, y)$ = number of times "`sunul bildir`" is seen in y

6.4. Results

In this section, first the baseline ASR systems and the experimental set-up for DLMs are explained. Then, the results of the DLM experiments with words and sub-lexical units are given for the proposed feature sets.

6.4.1. Baseline ASR systems

In the DLM experiments we used the baseline word and morph ASR systems with the liguistic segmentations as explained in Section 4.2. The WERs for the 1-best and the 50-best and 1000-best oracles are summarized in Table 6.1. Note that oracle WER is the lowest error rate that can be achieved with the N-best list reranking approaches.

Table 6.1. ASR results for the baseline systems

	WER (per cent)					
	Word system			Morph system		
	1-best	50-best oracle	1000-best oracle	1-best	50-best oracle	1000-best oracle
Held-out	24.1	15.4	12.4	22.9	14.2	11.4
Test	23.4	15.0	12.1	22.4	13.9	11.2

6.4.2. Experimental Set-up for DLMs

DLM training data, N-best lists, for word and morph recognition units were generated by decoding the acoustic model training data with the word and morph ASR systems. Language model over-training was controlled via 12-fold cross validation. Utterances in each fold were decoded with the baseline acoustic model trained on all the utterances and the fold-specific language model. 200 K word vocabulary and 76 K morph vocabulary were utilized in building 3-gram word and 4-gram morph language

models respectively. Note that the same vocabulary was utilized in building all the fold-specific language models. In the word model, the most frequent 200 K words in the generic text data and the reference transcriptions of the whole acoustic model training data were utilized as the vocabulary. The morph vocabulary contains all the morphs in the morph segmentations of the generic text data and the reference transcriptions of the acoustic model training data. The non-initial morphs in the morph segmentations were marked with "-" sign in order to find the word boundaries easily after recognition. A fold-specific language model was generated by interpolating the generic language model with the in-domain language model built from the reference transcriptions of the utterances in the other 11 folds. The same interpolation constant, 0.5, optimized for the baseline language model (generic language model interpolated with the in-domain language model built from the reference transcriptions of all the utterances in the acoustic model training data) was utilized in the fold-specific language models.

The perceptron algorithm, as presented in Section 6.1, was used for training the feature parameters. In our experiments oracle best path was used as the gold standard. 50-best and 1000-best word and morph hypotheses were utilized in discriminative training and reranking. In the perceptron model the parameter for the baseline model, α_0 (the weight associated with $\Phi_0(x, y)$), was set to a fixed number and was not updated during training. We trained 12 different models by changing the α_0 constant from 0 to 16. The best model was chosen using the held-out set WER. Figure 6.9 illustrates the optimization of α_0 parameter for word unigram features. For this feature set, the best improvement was obtained at $\alpha_0 = 2$. Averaged perceptron parameters were used in the evaluation of the held-out and test sets. The number of iterations over the training data were also optimized on the held-out set. We investigated the combinations of different n-gram orders (unigrams, unigrams+bigrams, unigrams+bigrams+trigrams) for the proposed feature sets. The n-gram combination yielding the lowest WER on the held-out set was chosen as the best combination of this feature set. If different combinations yielded the same amount of improvements, then the one with the lowest number of features was chosen as the best combination. Even though, more complicated scenarios that utilize the held-out set WER and the number of feature parameters in deciding

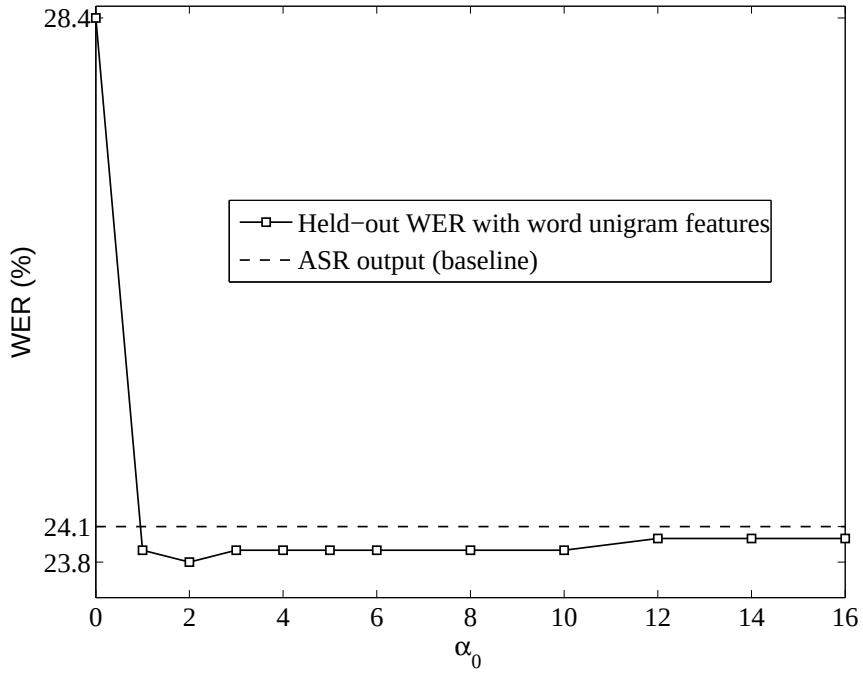


Figure 6.9. Effect of α_0 to DLM with words. Unigram word features are utilized for demonstration.

on the best n-gram feature combination are possible, we selected this simple approach in our experiments. Features from other information sources were incorporated into the best n-gram combination of the current feature set.

6.4.3. DLM Experiments with Words

This section presents the DLM experiments with the proposed feature sets explained in Section 6.2. Feature sets were extracted from the word DLM training data and utilized to rerank the word N-best lists of the held-out and the test utterances. Feature sets are represented with the notations listed in Table 6.2 in this section.

6.4.3.1. Experimental Results with Word n-gram Features. We first experimented with word n-gram features. The results for the held-out set are given in Tables 6.3 and 6.4. In Table 6.3, word n-gram features were extracted from the 50-best list of the training data and the 50-best lists of the held-out utterances were reranked with the feature parameters estimated with the perceptron algorithm. The same procedure was followed

Table 6.2. Notations and descriptions for the feature sets used in DLM experiments with words. Details of the feature sets are explained in Section 6.2.

Feature set	Notation	Description
Basic Features		
Word	$W(1)$	word unigrams
	$W(1,2)$	word unigrams+bigrams
	$W(1,2,3)$	word unigrams+bigrams+trigrams
Morphological Features		
Root	$R(1)$	root unigrams
	$R(1,2)$	root unigrams+bigrams
	$R(1,2,3)$	root unigrams+bigrams+trigrams
Stem+ending	$SE(1)$	stem+ending unigrams
	$SE(1,2)$	stem+ending unigrams+bigrams
	$SE(1,2,3)$	stem+ending unigrams+bigrams+trigrams
Inflectional Group	$IG(1)$	IG unigrams
	$IG(1,2)$	IG unigrams+bigrams
	$IG(1,2,3)$	IG unigrams+bigrams+trigrams
	$IG^{*}(2)$	IG bigrams (between IGs in consecutive words)
	$IG^{*}(2,3)$	IG bigrams+trigrams (between IGs in consecutive words)
	$IG^{**}(2)$	IG bigrams (between last IG of the current word and IGs of the next word)
Statistical Sub-lexical Features		
Morph	$M(1)$	Morph unigrams
	$M(1,2)$	Morph unigrams+bigrams
	$M(1,2,3)$	Morph unigrams+bigrams+trigrams
Syntactic Features		
PoS tag	$P(1)$	PoS unigrams
	$P(1,2)$	PoS unigrams+bigrams
	$P(1,2,3)$	PoS unigrams+bigrams+trigrams
Head-to-head	$H2H(w,w)$	H2H dependencies (between words)
	$H2H(w,t)$	H2H dependencies (between the dependent word and the PoS tag of the other word)
	$H2H(t,w)$	H2H dependencies (between the PoS tag of the dependent word and the other word)
	$H2H(t,t)$	H2H dependencies (between PoS tags)

Table 6.3. DLM results with word n-gram features. 50-best list is utilized in estimating the feature parameters and in reranking the held-out hypotheses.

Trials	Feats	ActFeats	held-out	
(on word-based system)	($\times 10^3$)	($\times 10^3$)	WER	ΔWER
ASR output	-	-	24.1	-
$W(1)$	154.9	51.2	23.8	0.3
$W(1,2)$	4037.9	330.8	23.8	0.3
$W(1,2,3)$	12087.4	931.1	23.9	0.2

Table 6.4. DLM results with word n-gram features. 1000-best list is utilized in estimating the feature parameters and in reranking the held-out hypotheses.

Trials	Feats	ActFeats	held-out	
(on word-based system)	($\times 10^3$)	($\times 10^3$)	WER	ΔWER
ASR output	-	-	24.1	-
$W(1)$	189.9	56.0	23.8	0.3
$W(1,2)$	19926.0	462.1	23.8	0.3
$W(1,2,3)$	72516.8	1152.6	23.8	0.3

with 1000-best lists and the results are given in Table 6.4. In these tables, "Feats" represents the number of features extracted from the N-best lists, 50-best or 1000-best lists. The number of the active features, features with non-zero weights after the parameter training, is denoted by "ActFeats".

The perceptron model trained on 50-best list yields 0.3 per cent improvement with unigram features over the baseline held-out set error. Incorporating higher order n-grams into the unigram feature set does not introduce any further gains. Unigram features also yield 0.4 per cent improvement (significant at $p = 0.009$) over the baseline test set error. When we experiment with 1000-best list, the number of features increases significantly, however, the same improvement is obtained with the model trained on 50-best list. Therefore, we only utilized 50-best lists in the experiments with the other feature sets.

Table 6.5. DLM results with root n-gram features.

Trials (on word-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	24.1	-
$R(1)$	34.3	12.9	23.4	0.7
$R(1,2)$	1778.5	235.6	23.7	0.4
$R(1,2,3)$	6738.8	657.2	23.7	0.4

Table 6.6. DLM results with stem+ending n-gram features.

Trials (on word-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	24.1	-
$SE(1)$	51.0	21.7	23.5	0.6
$SE(1,2)$	1991.7	311.2	23.6	0.5
$SE(1,2,3)$	8655.6	906.8	23.7	0.4

6.4.3.2. Experimental Results with Sub-lexical Features. Second we investigated sub-lexical features in DLMs. Root, stem+ending and IG-based n-grams were utilized as the morphological and morph n-grams were utilized as the statistical sub-lexical features.

The results of the **root n-gram features** are given in Table 6.5. Root unigram features yield the highest improvement on the held-out set error, resulting in also 0.4 per cent improvement (significant at $p = 0.002$) on the test set error. However, the root unigram features seem to over-train on the held-out set since the gain obtained on the held-out set is not preserved on the test-set.

The results of the **stem+ending n-gram features** are given in Table 6.6. Unigram stem+ending features yield the highest improvement on the held-out set error, resulting in 0.8 per cent improvement (significant at $p < 0.001$) on the test set error.

Table 6.7. DLM results with IG-based n-gram features.

Trials (on word-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	24.1	-
$IG(1)$	36.0	16.9	23.2	0.9
$IG(1,2)$	765.4	208.5	23.4	0.7
$IG(1,2,3)$	4716.7	747.9	23.4	0.7
$IG(1)+IG^{*}(2)$	3355.6	496.9	23.6	0.5
$IG(1)+IG^{*}(2,3)$	35535.0	4305.0	23.6	0.5
$IG(1)+IG^{**}(2)$	167.9	57.0	23.3	0.8

The results of the **IG-based n-gram features** are given in Table 6.7. The group of trials at the upper rows represent the experiments with the adjacent IG n-grams and the group of trials at the lower rows represent the experiments with n-grams between IGs of the consecutive words. From Table 6.7, it is clear that the number of n-grams obtained from consecutive words is much more than the number of adjacent n-grams. Most probably due to the sparsity of observations introduced with higher number of features, bigram and trigram features of consecutive words result in lower gains than adjacent bigram and trigram features. $IG(1)+IG^{**}(2)$ feature set decreases the number of features significantly compared to $IG(1)+IG^{*}(2)$ feature set by considering only the pairs that may convey the syntactic dependency relations between words as the bigram features. As a result $IG(1)+IG^{**}(2)$ feature set yields higher gain on the held-out set error than $IG(1)+IG^{*}(2)$ feature set, also yields 0.1 per cent higher gain on the test-set error. The best improvement on the held-out set error is obtained with IG unigram features, resulting in also 0.8 per cent improvement (significant at $p < 0.001$) on the test set error.

The results of the **morph n-gram features** are given in Table 6.8. As explained in Section 6.2.2.2, word hypothesis sentences were converted to morph hypothesis sentences by replacing each word with its morph segmentation obtained from the Morfessor algorithm. Then n-gram morph features were extracted and utilized to rerank the word

Table 6.8. DLM results with statistical morph n-gram features.

Trials (on word-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	24.1	-
$M(1)$	33.2	19.1	23.6	0.5
$M(1,2)$	2240.8	314.9	23.6	0.5
$M(1,2,3)$	9278.0	892.9	23.6	0.5

50-best hypotheses. Incorporating bigram and trigram features to the unigram feature set increases the number of features significantly, and bigram and trigram features do not introduce further improvements on the gain obtained with the unigram features. Unigrams yield the lowest number of features with the same amount of improvement on the held-out set and achieve also 0.5 per cent improvement (significant at $p < 0.001$) on the test set error.

6.4.3.3. Experimental Results with Syntactic Features. Third we investigated syntactic features, PoS tag n-grams and H2H dependencies, in reranking the word hypotheses. PoS tag features were applied on top of the word n-gram features and H2H dependency features were applied on top of the word and PoS tag n-gram features as additional information sources. Each additional feature set introduced information from higher order n-gram context. For instance, unigram+bigram and unigram+bigram+trigram PoS tag features were incorporated into word unigram features and unigram+bigram+trigram PoS tag features were incorporated into word unigram+bigram features. Additionally, H2H dependency relation features were incorporated into the set of word and PoS tag n-gram features.

The results of the experiments with word and PoS tag n-gram features are given in Table 6.9. PoS tag n-gram features introduce additional gains on top of the gains obtained with word n-gram features. PoS tag n-gram features yield the same amount of gains when they are incorporated into word unigram and word unigram+bigram features. $W(1) + P(1,2)$ feature set gives the lowest number of features, therefore

Table 6.9. DLM results with word and PoS tag n-gram features.

Trials (on word-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	24.1	-
$W(1)$	154.9	51.2	23.8	0.3
$W(1) + P(1,2)$	155.7	60.6	23.4	0.7
$W(1) + P(1,2,3)$	170.0	73.4	23.4	0.7
$W(1,2)$	4037.9	330.8	23.8	0.3
$W(1,2) + P(1,2,3)$	4053.0	552.8	23.4	0.7

it is selected as the best combination among the other feature combinations given in Table 6.9. This feature set also yields additional 0.5 per cent improvement (significant at $p < 0.001$) on top of the gain obtained with word unigram features on the test set.

The results of the experiments with word and PoS tag n-gram and H2H dependency relation features are given in Table 6.10. In this table, $H2H(all)$ represents utilizing all the H2H features together ($H2H(all) = H2H(w,w) + H2H(t,w) + H2H(w,t) + H2H(t,t)$). We only incorporate H2H dependency relation features to $W(1) + P(1,2)$ feature set. Incorporating $H2H(all)$ features to word and PoS tag n-gram features significantly increases the number features and most probably as a consequence of the sparsity of the observations, utilizing $H2H(all)$ features decreases the gain obtained

Table 6.10. DLM results with word and PoS tag n-gram and H2H dependency relation features.

Trials (on word-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	24.1	-
$W(1)$	154.9	51.2	23.8	0.3
$W(1) + P(1,2)$	155.7	60.6	23.4	0.7
$W(1) + P(1,2) + H2H(all)$	5783.6	1152.5	23.6	0.5
$W(1) + P(1,2) + H2H(tt)$	157.5	66.7	23.4	0.7

with the PoS tag n-gram features. Therefore, we only investigate $H2H(t,t)$ feature set as an additional source of information to word and PoS tag n-gram features. The motivation for selecting this feature set among the other H2H features is that PoS tag n-gram features are shown to be useful in Table 6.9 and $H2H(t,t)$ features convey additional information on PoS tags, the dependency relations between these units. However, we do not observe any additive gain.

To sum up, the best result is obtained with unigram IG features. Incorporating PoS tag unigram+bigram features into word unigrams gives the second best result. In order to understand whether there is any complementary information in the IG and the PoS tag features, we utilize these two features together ($IG(1)+P(1,2)$) in another DLM experiment. $IG(1)+P(1,2)$ feature set reduces the held-out set error from 24.1 per cent to 23.2 per cent and the test set error from 23.4 per cent to 22.4 per cent. This feature set yields slightly better test set gains than utilizing IG features alone and PoS tag features with word unigrams. IGs also contain PoS labels in their tag sequences, however, $P(1,2)$ feature set and $IG(1)$ feature set convey different tag factorizations – PoS tags incorporate lexical level and PoS labels in IGs incorporate sub-lexical level information.

6.4.4. DLM Experiments with Sub-lexical Units

This section presents the DLM experiments with the proposed feature sets explained in Section 6.3. Feature sets were extracted from the morph DLM training data and utilized to rerank the morph 50-best lists of the held-out and the test utterances. In this section, feature sets are represented with the notations listed in Table 6.11. We followed the same training procedure with the experiments given in Section 6.4.3.

6.4.4.1. Experimental Results with Morph n-gram Features. First we experimented with **morph n-gram features**. The results are given in Table 6.12. The best result on the held-out set error is obtained with the morph unigram features, yielding also 0.6 per cent improvement (significant at $p < 0.001$) on the test set error. As

Table 6.11. Notations and descriptions for the feature sets used in DLM experiments with sub-lexical units. Details of the feature sets are explained in Section 6.3.

Feature set	Notation	Description
Sub-lexical Features		
Morph	$M(1)$	morph unigrams
	$M(1,2)$	morph unigrams+bigrams
	$M(1,2,3)$	morph unigrams+bigrams+trigrams
	$M_{WI}(2)$	word internal morph bigrams
	$M_{FM}(1)$	first morph unigrams
	$M_{FM}(2)$	first morph bigrams
Morpho-syntactic Features		
Morph Clusters (with Brown et al.'s algorithm)	$C_B(1,2)$	Morph cluster unigrams+bigrams
Morph Clusters (with MED similarity)	$C_{MED}(1,2)$	Morph cluster unigrams+bigrams
Long Distance Triggers	$M_{LD}(2)$	Morph trigger pairs

Table 6.12. DLM results with morph n-gram features.

Trials (on morph-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	22.9	-
$M(1)$	45.9	20.1	22.1	0.8
$M(1,2)$	2272.7	241.3	22.4	0.5
$M(1,2,3)$	9242.1	826.5	22.3	0.6

with the other feature sets utilized in word DLM experiments, bigrams and trigrams do not introduce any further gains over unigram features. DLM yields more improvement for morphs than for words with basic n-gram features (Compare Table 6.3 with Table 6.12). This result demonstrates the superiority of sub-lexical units also in DLMs.

Then we utilized **word internal morph n-grams** to deal with non-word morph sequences and **first morph n-grams** to cover the word bigram context with morph

Table 6.13. DLM results with word internal and first-morph n-gram features.

Trials (on morph-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	22.9	-
$M(1)$	45.9	20.1	22.1	0.8
$M(1) + M_{WI}(2)$	435.3	76.7	22.4	0.5
$M(1)$	45.9	20.1	22.1	0.8
$M_{FM}(1)$	32.2	16.6	22.4	0.5
$M(1) + M_{FM}(2)$	2037.1	213.6	22.3	0.6
$M(1) + M_{WI}(2) + M_F(2)$	2426.5	245.1	22.4	0.5

bigrams. The results are given in Table 6.13. Incorporating neither word internal nor first morph bigrams to morph unigrams introduce additive improvements to the gain obtained with morph unigram features. Utilizing only first-morph unigrams results in lower gains than utilizing all the morph unigrams. If we make an analogy between roots and first morphs and between suffixes and non-initial morphs, this finding is consistent with the one obtained in Section 6.4.3.2 where stem+ending unigram features outperform root unigram features (Compare Table 6.5 with Table 6.6). We also utilize morph unigrams, first-morph and word internal morph bigrams together, however, we do not obtain any improvement over the gain of morph unigram features.

6.4.4.2. Experimental Results with Morpho-syntactic Features. Statistical morphs do not convey any linguistic information like grammatical morphemes. Therefore morpho-syntactic features were proposed to reveal the implicit syntactic information on morph sequences. Morpho-syntactic features were utilized together with morph unigram features in the experiments.

Before reporting the performance of the morpho-syntactic features, we investigated the most obvious way of obtaining syntactic features on morph sequences. This approach consists of converting morph sequences into word-like units and utilizing the linguistic tools on these units to obtain syntactic features. Here, it is important to note

Table 6.14. DLM results with morph and PoS tag n-gram features.

Trials (on morph-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	22.9	-
$M(1)$	45.9	20.1	22.1	0.8
$M(1) + P(1,2)$	46.7	22.2	21.7	1.2

that not all of the concatenated morph sequences are grammatically correct Turkish words. In the morphological analyzer, these non-word sequences were also analyzed if they end with grammatically correct suffix concatenations. Otherwise, they were left as unparsed and represented as nominal nouns. After processing the word-like units with the linguistic tools, the same procedure with words was applied to extract the syntactic features and to rerank the morph hypotheses. The feature definitions are the same with the definitions in Section 6.2.3. We only utilized PoS tag n-grams as the syntactic features. This approach resulted in using PoS tag n-gram features of words together with morph unigram features. The result of this feature set is given in Table 6.14. PoS tag unigram+bigram features yield 0.4 per cent additive improvement (significant at $p < 0.001$) on top of the gain obtained with morph unigram features both on held-out and test sets.

As explained in Section 6.3.2, morph clusters and long distance morph triggers were utilized in defining the morpho-syntactic features. Cluster labels of morphs were analogous to PoS tags of words and morph trigger pairs were analogous to H2H dependency relations.

In clustering with Brown et al.'s algorithm, we performed clustering trials with three different predefined number of clusters (50, 100 and 200 clusters). Each trial gave almost the same improvement. We preferred to report the result of only the trial with 50 clusters since this trial provided the smallest number of features. In clustering of non-initial morphs with MED similarity, we constrained the hierarchical clustering algorithm by restricting the merges if similarity of two clusters is less than a threshold that is proportional to the lengths of the most distant string pairs. This thresholding

Table 6.15. DLM results with morph and PoS tag n-gram features.

Trials (on morph-based system)	Feats ($\times 10^3$)	ActFeats ($\times 10^3$)	held-out WER	held-out ΔWER
ASR output	-	-	22.9	-
$M(1)$	45.9	20.1	22.1	0.8
$M(1) + C_B(1,2)$	48.5	24.2	21.8	1.1
$M(1) + C_{MED}(1,2)$	95.1	28.5	21.8	1.1
$M(1) + M_{LD}(2)$	46.9	21.3	22.2	0.7
$M(1) + C_B(1,2) + M_{LD}(2)$	49.5	27.0	21.8	1.1
$M(1) + C_{MED}(1,2) + M_{LD}(2)$	96.1	28.0	21.8	1.1

approach prevented the merges of clusters that contain shorter strings. Hierarchical clustering of non-initial morphs with MED similarity resulted in 5185 clusters and all the initial morphs were put into the same cluster. The results of these feature sets are given in Table 6.15. Both of the clustering approaches yield the same amount of improvement on the held-out set and result in 0.5 per cent (significant at $p < 0.001$) additional gain over the test set error achieved with the morph unigram features.

Then we investigated long distance trigger features. The candidate long distance pairs were obtained from the 1-best and oracle hypotheses of the morph 50-best lists and this resulted in 6.8 M candidate pairs. Rejecting the null hypothesis (independence assumption of the morphs in the pairs) at the confidence level of 0.001 resulted in 1.32 M long distance trigger features. In order to decrease the number of the trigger features, we sorted the pairs in the descending order according to their log likelihood ratios and chose only the pairs with the highest log likelihood ratios. We performed trials with the trigger pairs with the highest 1 K, 10 K, 50 K and 100 K log likelihood ratios. These trigger pairs were incorporated into the morph unigram features and into the combination of morph unigram and morpho-syntactic unigram+bigram features. We did not obtain any significant differences in the performances of different trials. Therefore, we only report the results of the trial with 1 K long distance trigger features in Table 6.15. Incorporating this feature set in neither morph unigram features nor morph unigram and morpho-syntactic unigram+bigram features yield any

improvements.

To sum up, the best result on the held-out set error is obtained with $M(1)+P(1,2)$ feature set. The morpho-syntactic features together with morph unigram features yield almost the same amount of improvements with this best scoring feature set (Compare Table 6.14 with Table 6.15). It is very promising for the clustering approaches to achieve almost the same results with the exact PoS tag clusters of concatenated morph sequences. Unfortunately, long distance trigger features do not give any improvements.

6.5. Analysis of the Results

The DLM experiments on both word and morph hypotheses reveal some important and interesting results. First, basic unigrams are shown to be effective for obtaining significant gains on the baseline and increasing the n-gram context does not introduce any further gain. Considering that the perceptron training penalizes features associated with the current 1-best and rewards features associated with the oracle, the success of the word unigram features can be explained as adaptation of the language model with the perceptron algorithm [119]. Language model adaptation is important in generative language models to obtain robust n-gram estimates since the in-domain text data (reference transcriptions of the Broadcast News recordings) is very small compared to the generic text corpus (newspaper articles). In order to alleviate the effect of out-of-domain data, generic and in-domain language models are linearly interpolated and the language model interpolation constant is optimized on the held-out set before lattice generation (See Table 3.2 for the effect of the linear interpolation on the ASR performance). However, DLM with unigram features can provide an extra adaptation by biasing the words or morphs occurring more frequently in the in-domain data.

In order to investigate if the gains of the word and morph unigram features are really coming from language model adaptation, we investigate the correlation between the active unigram feature weights and the difference of the unigram log probabilities between the in-domain and the baseline language models both for words and morphs. Observing a higher correlation between these two component will be a strong evidence

Table 6.16. Summary of the DLM results for unigram sub-lexical features.

Trials (on word-based system)	Feats ($x10^3$)	ActFeats (*percent*)	held-out ΔWER	test ΔWER
$R(1)$	34.3	12.9	0.7	0.4
$M(1)$	33.2	19.1	0.5	0.5
$SE(1)$	51.0	21.7	0.6	0.8
$IG(1)$	36.0	16.9	0.9	0.8

of the language model adaptation. However, we obtain lower correlations, 0.09 for words and 0.16 for morphs, which means that the gain obtained with the word and morph unigram features is not only coming from the language model adaptation.

Second, we demonstrate the superiority of sub-lexical units also in DLMs in addition to generative language models. When we compare reranking word hypotheses with basic word features and reranking morph hypotheses with basic morph features, morphs yield more improvement than words. The gain obtained with morph unigrams, 0.8 per cent on held-out and 0.6 per cent on test, is higher than the gain obtained with word unigrams, 0.3 per cent on held-out and 0.4 per cent on test. Additionally, sub-lexical units also outperform words when reranking word hypotheses. Table 6.16 summarizes the performance of the sub-lexical unigram features utilized in the DLM experiments on word hypotheses. Morph and IG features seem to be more robust than the other features, since the gains observed in held-out set are preserved in the test set. Additionally, root unigram features seem to overtrain on the held-out data. Among the sub-lexical units given in Table 6.16, IGs convey more linguistic information than stem+endings and stem+endings convey more linguistic information than roots. Even though morphs do not carry any explicit linguistic information, intuitively we can say that morphs convey more information than roots since morph unigram features utilize both initial and non-initial morphs. Except for root unigrams, the amount of the linguistic information conveyed in the sub-lexical features is reflected to the gain obtained on the baseline error. In terms of the test set error, IGs and stem+endings perform significantly better than roots (significant at $p = 0.007$ and $p = 0.002$ respectively).

Table 6.17. Summary of the DLM results for PoS tag and morpho-syntactic cluster features.

Trials (on word-based system)	Feats ($x10^3$)	ActFeats (*percent*)	held-out ΔWER	test ΔWER
$W(1)$	154.9	51.2	0.3	0.4
$W(1)+P(1,2)$	155.7	60.6	0.7	0.9
Trials (on morph-based system)				
$M(1)$	45.9	20.1	0.8	0.6
$M(1)+P(1,2)$	46.7	22.2	1.2	1.0
$M(1)+C_B(1,2)$	48.5	24.2	1.1	1.1
$M(1)+C_{MED}(1,2)$	95.1	28.5	1.1	1.1

The 0.3 per cent improvement obtained over the gain of morphs both with *IGs* and stem+endings is not statistically significant.

Third, it is shown that the n-gram features that capture the generalizations of the training data are effective in DLM experiments. PoS tags are obtained with linguistic tools and they capture the syntactic groupings of words and word-like units (concatenated morph sequences). Morpho-syntactic clusters try to group morphs according to common syntactic functions and the clustering of morphs is learned with data-driven approaches on morph sequences. PoS tags and morpho-syntactic clusters achieve significant amount of additive gains on top of the gains obtained with word and morph unigram features. Table 6.17 summarizes the results with these feature sets. Utilizing PoS tags alone ($P(1,2)$ feature set only) does not yield any improvement over the baseline error and utilizing morpho-syntactic clusters ($C_B(1,2)$ feature set only) yield 0.3 per cent significant (at $p < 0.001$) improvement over the baseline test set error. It is interesting that the gains obtained with combining basic n-gram features with PoS tag or morpho-syntactic features are independent of the clustering approach.

Fourth, neither H2H dependency relation nor morph trigger features give any improvements. Morph triggers are just a brute-force attempt to incorporate longer

distance morph relations into DLMs and they fail as DLM features. However, H2H dependencies are linguistically motivated, therefore they are expected to be more effective in DLMs. Section 4.3.2 reported the error analysis of the best scoring baseline system and we found out that 99 per cent of the errors labelled as correctable are coming from syntactic errors. H2H dependency features are proposed to correct the syntactic errors in hypothesis sentences, but they do not introduce any gains in the DLM experiments. $H2H(all)$ features even degraded the performance of the existing feature set when they are incorporated into the combination of word and PoS tag n-gram features. One possible problem is that the hypothesis sentences in the DLM training data contain recognition errors and the parser generates the best possible dependency relation even for incorrect hypotheses. As a result, the dependency analysis of the incorrect hypotheses may not provide good negative examples for discrimination of the correct and incorrect hypotheses.

Another possible problem is the sparseness of the observations per parameters. If a feature is not seen frequently enough in the training data, we can not estimate a robust parameter for this feature with the perceptron training. Additionally, parameter updates for infrequent features affect also the parameter updates for the frequent features. As a result, incorporating sparse features can degrade the performance of the existing model. In generative language modeling, higher order n-grams have been shown to be more effective than unigrams. However, in discriminative language modeling, utilizing higher order n-grams does not give any improvements. Data sparseness can explain the reason of this incompatible result in DLMs since bigrams and trigrams introduce too many features to the model compared to the unigram features. We believe that the high number of features are masking the expected gains of the proposed features. This will make feature selection a crucial issue in our future research.

Finally, we investigate the gains of the best scoring feature sets in word and morph DLM experiments according to the background conditions. The classical Hub4 classes are used in the background conditions: (f0) clean speech, (f1) spontaneous speech, (f2) telephone speech, (f3) background music, (f4) degraded acoustic conditions, and (fx) all other speech. The baseline systems yield the lowest WER for the clean speech (See

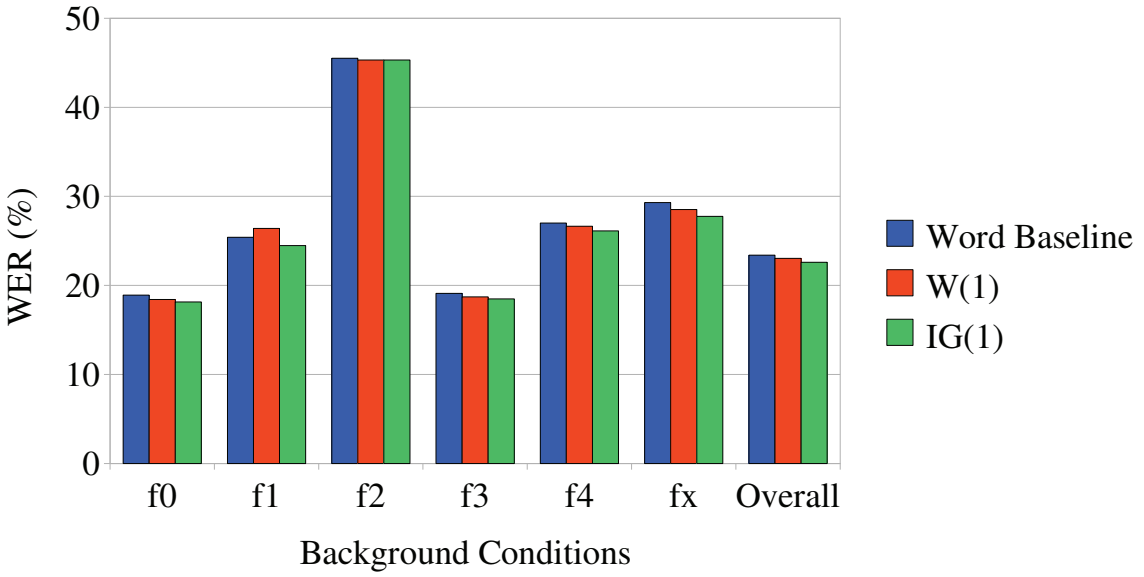


Figure 6.10. Comparison of WERs for the word baseline system, DLM with basic word unigram features ($W(1)$) and DLM with the best scoring feature set ($IG(1)$) according to background conditions.

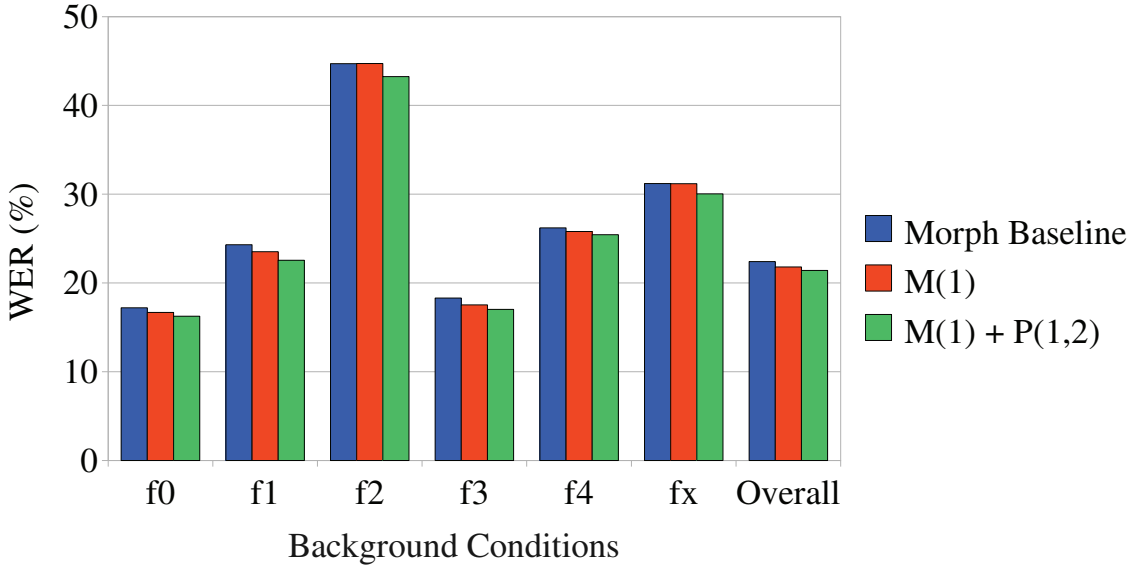


Figure 6.11. Comparison of WERs for the morph baseline system, DLM with basic morph unigram features ($M(1)$) and DLM with the best scoring feature set ($M(1) + P(1,2)$) according to background conditions.

Figure 4.2 for the comparison of the word and the morph baselines only). Figures 6.10 and 6.11 illustrate the comparison of WERs according to the background conditions for the DLM experiments with word and morph hypotheses respectively. The WERs for the baseline systems, the WERs for the DLM experiments with basic unigram features ($W(1)$ and $M(1)$) and the WERs for the DLM experiments with the best scoring feature sets ($IG(1)$ for words and $M(1)+P(1,2)$ for morphs) are given in the Figures. DLM with the basic unigram features outperform the baseline systems in most of the background conditions and the best scoring feature sets outperform the basic unigram features in all of the background conditions. The lowest WER obtained on the clean speech (f0) with DLMs is 16.3%.

7. CONCLUSIONS

In this dissertation, we address the challenges of Turkish for LVCSR systems with novel language modeling approaches. First, we develop two ASR systems, one for newspaper content transcription and the other for broadcast news transcription. Then we investigate the effectiveness of our proposed approaches on these systems. Following sections summarize the main conclusions extracted from each proposed approach.

7.1. Sub-lexical Units for Language Modeling

High OOV rates and non-robust language model estimates are the main challenges of Turkish as well as other morphologically rich languages in LVCSR systems. Sub-lexical units for language modeling have been proposed to handle these challenges for morphologically rich languages in the literature. We also investigate sub-lexical language modeling units in Turkish ASR. Grammatical sub-lexical units, obtained with morphological analyzers, and statistical sub-lexical units, obtained with the Morfessor algorithm, are explored for Turkish LVCSR. We also compare the performances of sub-lexical units with very large vocabulary word language models.

Our experiments on the BN transcription system show the superiority of sub-lexical units in modeling Turkish language. We find that our best scoring sub-lexical model, morph-based model, performs significantly better than the word-based model even for very large vocabulary sizes. The morph-based model and the cheating experiment that investigates the effect of handling OOV words by incorporating all the OOV words in the test data to the recognition vocabulary yield the same performance. This finding is a strong evidence of the superiority of morphs over words due to handling the OOV problem. We also observe that the WER differences between the best scoring sub-lexical model and the word models are getting smaller with the increasing vocabulary coverage in the word models. This shows that OOV problem can also be addressed by using very large vocabularies when huge language model training corpora are available. In our experiments, the baseline morph model outperforms the word model even

with 500 K vocabulary by 0.8 per cent (significant at $p < 0.001$). This suggests larger vocabulary word-based ASR systems for Turkish may reach the same performance with the morph-based ASR system. However, it may not be possible to accommodate very large vocabularies and training data due to computational limitations, since very large vocabularies require larger training data for robust language model estimates. Therefore, sub-lexical units are the best way of handling the challenges of Turkish in LVCSR systems with moderate size vocabularies.

We also analyze the recogniton errors of the morph-based system. First, we explore if the gains obtained with sub-lexical approaches are really coming from dealing with OOV words. We find out that 1.0 per cent of the gain obtained with the morph-based system over the baseline word model is mostly coming from recognising the OOV words correctly. This finding also pronounces the superiority of morphs over words due to handling the OOV problem. Second, we explore the type of the recognition errors, however, this analysis is performed only on the 6.4 per cent out of 22.4 per cent recognition errors. The main conclusion extracted from this analysis is that there is a possibility of correcting almost 1.2 per cent of the recognition errors, out of 4.0 per cent, with reranking approaches that address the syntactic errors. The finding leads us to investigate discriminative language models with syntactic features.

7.2. Lattice Extension and Dynamic Vocabulary Adaptation

We also investigate dealing with the OOV problem directly on the word-based system with lattice extension and dynamic vocabulary adaptation techniques. In these approaches, the new words that will be added to the baseline vocabulary are learned using the first-pass lattice output of the baseline system with the assumption that OOV words are replaced by acoustically similar IV words during decoding. The performance of these approaches are investigated on the newspaper content transcription system.

Lattice extension and vocabulary adaptation approaches yield significant accuracy improvements over the baseline word model. However, predetermined vocabularies perform better than these techniques for huge vocabulary sizes. The main reason of this

result is the inclusion of the rare words to the adapted vocabularies which makes data sparseness more crucial for huge vocabulary sizes. The effectiveness of the adapted vocabularies over predetermined ones is more pronounced for moderate vocabulary sizes. The analysis of the improvements obtained with the lattice extension and vocabulary adaptation approaches reveals that the gains are coming from better OOV handling which is the main motivation of the proposed techniques.

7.3. Lattice Extension for Sub-lexical Units

In both of our ASR systems, the OOV problem is best handled by using sub-lexical language modeling units. Although, using sub-lexical units alleviates the OOV problem, sub-lexical units may result in ungrammatical items since the ASR system can generate any combination of sub-lexical units which include non-word items. In the BN transcription system, the morph-based model introduces 0.4 per cent recognition errors due to over-generated items and in the newspaper content transcription system the morph-based model introduces 2.0 per cent recognition errors due to over-generated items. This shortcoming of the morph-based system leads us to address the over-generation problem of sub-lexical units.

In this dissertation, we deal with the over-generation problem with a second-pass approach on the morph-based system. Over-generated units can be corrected with simple morphological constraints if the sub-lexical units convey morphological features. Since statistical morphs do not carry explicit linguistic information, the lattice extension strategy is modified to map morph sequences to grammatically correct Turkish words. Our proposed approach improves the performance of the morph-based model in the newspaper content transcription system by 1.6 per cent absolute in WER by mostly covering the recognition errors due to over-generation.

7.4. Discriminative Language Models with Linguistically and Statistically Motivated Features

We investigate linguistically and syntactically motivated features in addition to the basic word n-gram features in Turkish DLMs. The linguistically motivated features are extracted from the morphological and syntactic information with the help of morphological and dependency parsers. Statistically motivated features are extracted from the morph sequences. Morph cluster n-gram and morph trigger features are proposed to reveal the implicit morpho-syntactic information conveyed by morphs.

The DLM experiments with words and morphs are performed on the BN transcription system. In the DLM experiments with words, the best result is obtained with the morphological feature set, IG-based n-grams. In the DLM experiments with morphs, the best result is obtained with the integration of morph n-gram features with PoS tag n-gram features of word-like units obtained from morph sequences. The morpho-syntactic clusters yield almost the same amount of gain with PoS tags of word-like units. It is interesting that the gain obtained with morpho-syntactic features is independent of the clustering approach, showing the effectiveness of the n-gram features that capture the generalizations of the training data. Neither H2H nor morph triggers give any improvements. Morph triggers are just a brute-force attempt to incorporate longer distance morph relations. However, H2H dependencies are linguistically motivated, therefore they are expected to be more effective in DLM. One possible problem is that the parser generates the best possible dependency relations even for incorrect hypotheses, as a result, it may not provide good negative examples for discrimination. Therefore, we will investigate a better way of incorporating longer distance information into DLM in our future research.

In all the proposed feature sets, unigrams are shown to be effective for obtaining significant gains on the baseline and increasing the n-gram context does not introduce any further gain. This finding leads us to think that this gain may come from adaptation of the language model with the perceptron algorithm. However, our investigations show that the gain is not coming only from language model adaptation.

Additionally, it is shown that DLM with basic morph n-gram features on morph hypothesis yield more improvement than DLM with basic word n-gram features on word hypothesis. This demonstrates the superiority of sub-lexical units also in DLM.

Our final observation is that the high number of features are masking the expected gains of the proposed features, mostly due to the sparseness of the observations per parameter. This will make feature selection a crucial issue for our future research.

7.5. Future Work

This dissertation reports promising results on Turkish LVCSR. However, further improvements can be possible with the below proposed approaches.

(i) Investigating improved statistical units for Turkish: The statistical sub-lexical units, obtained with the Morfessor algorithm, result in the best performance in Turkish ASR. However, further improvements can be performed on the Morfessor algorithm to obtain improved segmentations that result in better accuracies by taking the language characteristics of Turkish into account.

- All the affixes, except a few instances of prefixation, are suffixes in Turkish. Therefore, we make an analogy between initial morphs and roots and the non-initial morphs and suffixes in our research. However, there can be very short initial morphs in the segmentations, e.g., "a -nda". Most probably, these initial morphs are considered as prefixes during segmentation. Therefore, a length constraint derived from the root length distribution of Turkish, can be imposed to the algorithm to obtain longer initial morph segments.
- Even though we could not report the results of lexical grammatical units in Table 4.1, they have been shown to outperform statistical morphs in our previous research [2]. The success of these units are explained with two different reasons. First, they result in more robust n-gram estimates by considering surface form endings having the same lexical form representations as the same units in language modeling. Second, they handle the over-generation problem caused by incorrect morphophonemics. Considering these positive

effects of lexical form units, the algorithm can be modified to also find the lexical form representations of different surface form non-initial morphs.

The proposed algorithms in Morpho Challenge, that cover some of these properties, can be directly applied to Turkish or can be modified to take the characteristics of Turkish into account.

(ii) Feature selection and topic adaptation in DLMs: We investigate several feature sets for DLMs. The best improvement is obtained as absolutely 1.2 per cent on the held-out and 1.0 per cent on the test sets. In DLMs, we observe that having large number of features is masking the expected gains with the proposed feature sets. Therefore, feature selection will be a crucial future research.

- First, feature selection can be investigated on a given set of features. For instance, in word unigram features, only one third of the features became active after parameter training. Therefore, feature selection algorithms can be utilized to select only the useful features before parameter estimation.
- Second, feature selection and feature combination can be performed together. For instance, H2H feature set introduced around 5.7 M features to the word and PoS tag features. We thought that there can be complementary information between H2H, PoS and word features. However, H2H features did not yield any improvement, most probably, due to the sparsity of the features per parameter. It might be possible to reveal the complementary information between different feature sets by performing feature combination together with feature selection.
- Third, topic adaptation in DLMs can be performed. The motivation in this approach is that a set of features can be estimated more robustly in a specific topic or context than the others. A similar approach presented in [64] for ME models can be extended to DLMs to perform the topic adaptation.

(iii) Combining lattice extension and DLM for morphs: In this dissertation, statistical sub-lexical units, morphs, result in the best recognition accuracy. Then, lattice extension approach for morphs handles the over-generation problem of morphs and yields further improvements on the baseline morph result. In the DLM experiments, utilizing morph features on word hypotheses or directly on morph hypotheses yield significant improvements on the baseline. Therefore, an apparent

future work can be to combine lattice extension for morphs with DLM to obtain further gains.

APPENDIX A: DERIVATION OF EQUATION 6.7 FROM EQUATION 6.4

Notation	Meaning
T	number of words in the training data.
c_i	cluster of the i'th word w_i.
$C(w_1, w_2)$	number of occurrences of word bigram pair $w_1 w_2$
$C(c_1, c_2)$	number of occurrences of cluster bigram pair $c_1 c_2$
$\sum_{w_1 w_2}$	summation over all word bigram pairs
$\sum_{c_1 c_2}$	summation over all cluster bigram pairs
$H(w)$	entropy of the unigram word distribution
$I(c_1, c_2)$	average mutual information of adjacent classes

$L(\pi)$ is the log likelihood of the training data and defined as follows in [58].

$$L(\pi) = \frac{1}{(T-1)} \log P(w_2^T | w_1) \tag{A.1}$$

Using the Chain rule with bigram approximations

$$L(\pi) = \frac{1}{(T-1)} \sum_{i=2}^{i=T} \log P(w_i | w_{i-1}) \tag{A.2}$$

Using Equation 6.3,

$$L(\pi) = \frac{1}{(T-1)} \sum_{i=2}^{i=T} \log P(w_i | c_i) P(c_i | c_{i-1}) \tag{A.3}$$

$$\begin{aligned} L(\pi) &= \sum_{w_1 w_2} \frac{C(w_1 w_2)}{(T-1)} \log P(w_2 | c_2) P(c_2 | c_1) \\ &= \sum_{w_1 w_2} \frac{C(w_1 w_2)}{(T-1)} \log P(c_2 | c_1) + \sum_{w_1 w_2} \frac{C(w_1 w_2)}{(T-1)} \log P(w_2 | c_2) \end{aligned} \tag{A.4}$$

Equation A.4 is reorganized as follows:

$$
\begin{aligned}
L(\pi) &= \sum_{c_1 c_2} \frac{C(c_1 c_2)}{(T-1)} \log \frac{P(c_2|c_1)}{P(c_2)} + \sum_{w_1 w_2} \frac{C(w_1 w_2)}{(T-1)} \log \underbrace{P(w_2|c_2)P(c_2)}_{P(w_2)} \\
&= \sum_{c_1 c_2} P(c_1 c_2) \log \frac{P(c_2|c_1)}{P(c_2)} + \sum_{w_2} \frac{\sum_{w_1} C(w_1 w_2)}{(T-1)} \log P(w_2) \\
&= \sum_{c_1 c_2} P(c_1 c_2) \log \frac{P(c_1 c2)}{P(c_1)P(c_2)} + \sum_{w_2} \frac{C(w_2)}{(T-1)} \log P(w_2) \\
&= \underbrace{\sum_{c_1 c_2} P(c_1 c_2) \log \frac{P(c_1 c2)}{P(c_1)P(c_2)}}_{I(c_1,c_2):average\ \ mutual\ \ information} + \underbrace{\sum_{w} P(w) \log P(w)}_{-H(w):entropy} \qquad \text{(A.5)}
\end{aligned}
$$

Finally

$$
L(\pi) = I(c_1, c_2) - H(w) \qquad \text{(A.6)}
$$

APPENDIX B: DISTRIBUTION OF INITIAL AND NON-INITIAL MORPHS

Table B.1. Distribution of initial morphs (IM) and non-initial morphs (NIM) in 50 classes. Only the most probable members of the classes are given in the examples.

Class no	# of IM	# of NIM	Examples
1	12354	5042	i, di, al, tarafında, ardında, özellik, ...
2	12335	4121	kendi, yüzde, son, ilk, yer, orta, ...
3	3587	3378	-m, var, değil, -di, dedi, söyledi, ...
4	3347	2910	için, gibi, konusu, içinde, arasında, nedeni, ...
5	3158	1656	on, yirmi, otuz, elli, doksan, kırk, ...
6	2502	391	ve, yapan, veren, karşısında, arasındaki, yetkilileri, ...
7	2047	249	a, büyük, ya, sa, türkiye', yan, ...
8	1997	2975	-'ın, -'in, -'nin, -'a, -'un, -'de, ...
9	1318	1966	yıl, ülke, türk, dünya, türkiye, kişi, ...
10	928	165	de, -ce, bile, şekilde, den, hep, ...
11	680	1291	ne, yeni, e, iyi, avrupa, önemli, ...
12	659	19	olarak, olduğunu, olduğu, olma, aç, yapı, ...
13	654	19	sonra, kadar, karşı, göre, önce, zaman, ...
14	627	33	olan, ama, -ken, ancak, diye, ki, ...
15	647	24	başkanı, -en, genel, -erek, ilgili, bakanı, ...
16	233	805	-lı, -z, -dı, -dığı, -yor, -mış, ...
17	229	212	da, -ki, belirt, kayded, ver, itibar, ...
18	166	149	iki, dolar, lira, yaklaş, do, yıllık, ...
19	144	25	bir, binlerce, ikisi, toplu, ticari, yürür, ...
20	143	40	çok, daha, en, hiç, pek, teknik, ...
21	123	136	bin, sıfır, milyon, milyar, trilyon, mayıs, aralık, eylül, ocak, ...
22	79	130	-e, -er, -ine, -lığa, -liğe, araya, ...
23	71	284	-n, -cağı, -bileceği, -rak, -bilmesi, -mayan, ...
24	65	2	bu, o, her, türkiye'nin, istanbul, aynı, ...
25	61	1	yüz, nokta, yılı, saat, yılında, maçta, ...
26	47	12	-k, etmek, etmiş, etmesi, etmeye, edilecek, ...
27	45	24	-den, -ca, -ü, birliği, -yse, -ün, ...
28	39	157	-lar, -uz, -um, -du, okudu, mu, ...
29	33	27	beş, üç, dört, dokuz, altı, yedi, ...
30	12	99	-a, -s, -t, -ma, -un, -ına, ...
31	9	68	-si, -sini, -sinin, -diği, -miş, -siyle, ...
32	8	103	-i, -ini, -inin, -iyle, -ince, -inden, ...
33	7	38	-ler, -lik, -sin, -niz, -im, -ak, ...
34	7	13	-nin, -yle, -miz, edilen, -mi, -nüz, ...
35	6	15	-ları, -larını, -ını, -unu, -yu, -sız, ...
36	3	26	-ne, -nde, -nden, -ndeki, dakikada, -'ne, ...
37	3	25	-li, -in, -te, -ten, -inde, -ğı, ...
38	2	37	-ı, -'nın, -'da, -lık, -ın, -oğlu, ...
39	2	15	-da, -larda, -sında, -larında, -daki, -ında, ...
40	1	51	-ni, -u, -dığını, -nı, -diğini, belirtil, ...
41	1	36	-dan, -la, -ta, -p, -tan, -lu, ...
42	1	33	-le, -se, -arak, -yen, -p, -yerek, ...
43	1	22	-ya, -sına, -sından, -ktan, -ka, -kten, ...
44	1	13	-ye, -lere, -yi, -sine, -lerine, -si'ne, ...
45	1	11	-sı, -sını, -yı, -sının, -sıyla, -ması, ...
46	0	38	-nın, -yla, -mız, -nun, -'nun, -'nda, ...
47	0	13	-leri, -lerin, -lerini, -lerinin, -lerle, -lerden, ...
48	0	13	-ların, -lara, -larının, -larına, -lardan, -larla, ...
49	0	11	-de, -lerde, -sinde, -lerinde, -deki, -sel, ...
50	0	6	-ndan, -'ndan, -na, -nda, -na, -ndaki

APPENDIX C: MORPH TRIGGER PAIRS

Table C.1. Morph trigger pairs with the highest 35 log-likelihood ratios.

Morph trigger pairs	$\log(L(H_1)/L(H_0))$	Example morph sequences containing the triggers
iki → bin	21274.6	**iki bin** dokuz
avrupa → birliği	21186.6	**avrupa birliği** üyeliği
sayın → seyirci	19416.1	**sayın seyirci** -ler
parça → bulut	15783.8	**parça** -lı **bulut** -lu
-miş → milletler	15268.8	birleş **-miş milletler**
parça → -lu	14961.0	**parça** -lı bulut **-lu**
başbakan → erdoğan	14897.7	**başbakan erdoğan**
birleş → milletler	13983.1	**birleş** -miş **milletler**
sıcaklık → derece	11231.5	**sıcaklık** on **derece**
dışişler → bakanı	10782.3	**dışişler** -i **bakanı**
bin → yüz	9683.2	**bin yüz** on
-lı → bulut	9590.0	parça **-lı bulut** -lu
recep → tayyip	9070.4	başbakan **recep tayyip** erdoğan
-lı → -lu	8853.8	parça **-lı** bulut **-lu**
şu → a	8706.4	**şu a** -nda
sayın → -ler	8203.0	**sayın** seyirci **-ler**
hrant → dink	8097.1	**hrant dink**
sevgili → seyirci	7908.4	**sevgili seyirci** -ler
devam → ediyor	7890.2	yayını -mız **devam ediyor**
bu → arada	7244.5	**bu arada**
bin → dokuz	7100.0	iki **bin dokuz** yılında
bin → yılında	7014.0	iki **bin** dokuz **yılında**
tayyip → erdoğan	6963.4	başbakan recep **tayyip erdoğan**
diye → konuştu	6899.0	**diye konuştu**
amerika → birleş	6431.4	**amerika birleş** -ik devletleri
abdullah → gül	6290.2	cumhurbaşkanı **abdullah gül**
başbakan → tayyip	6213.0	**başbakan** recep **tayyip** erdoğan
recep → erdoğan	6207.5	başbakan **recep** tayyip **erdoğan**
on → beş	6023.0	**on beş**
başbakan → recep	6000.5	**başbakan recep** tayyip erdoğan
dokuz → yüz	5810.3	**dokuz yüz** on beş
şu → -nda	5746.6	**şu** a **-nda**
iç → kesim	5666.3	**iç kesim** -lerde
saddam → hüseyin	5607.9	**saddam hüseyin**
on → derece	5596.1	sıcaklık **on derece**

REFERENCES

1. Arısoy, E., D. Can, S. Parlak, H. Sak, and M. Saraçlar, "Turkish Broadcast News Transcription and Retrieval", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 17, No. 5, pp. 874 – 883, 2009.

2. Arısoy, E., H. Sak, and M. Saraçlar, "Language Modeling for Automatic Turkish Broadcast News Transcription", *Proceedings of Interspeech-Eurospeech*, pp. 2381 – 2384, Antwerp, Belgium, 2007.

3. Arısoy, E. and M. Saraçlar, "Analysis of the Recognition Errors in LVCSR of Turkish", *Proceedings of SIU*, pp. 361 – 364, Antalya, Turkey, April 2009.

4. Arısoy, E. and M. Saraçlar, "Lattice Extension and Vocabulary Adaptation for Turkish LVCSR", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 17, No. 1, 2009.

5. Arısoy, E., B. Roark, I. Shafran, and M. Saraçlar, "Discriminative N-gram Language Modeling for Turkish", *Proceedings of Interspeech*, pp. 825 – 828, Brisbane, Australia, 2008.

6. Arısoy, E., M. Saraçlar, B. Roark, and I. Shafran, "Syntactic and Sub-lexical Features for Turkish Discriminative Language Models", *Proceedings of ICASSP*, Dallas, Texas, USA, 2010.

7. Jurafsky, D. and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, New Jersey, 2000.

8. Jelinek, F., *Statistical Methods for Speech Recognition*, The MIT Press, 1997.

9. Huang, X., A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to*

Theory, Algorithm and System Development, Prentice Hall PTR, Upper Saddle River, New Jersey, USA, 2001.

10. Rabiner, L., "A Tutorial on HMM and Selected Applications in Speech Recognition", *Proceedings of IEEE*, Vol. 77, No. 2, pp. 257 – 286, February 1989.

11. Levenshtein, V., "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals", *Soviet Physics—Doklady*, Vol. 10, No. 10, pp. 707 – 710, 1966.

12. Aksungurlu, T., S. Parlak, H. Sak, and M. Saraçlar, "Comparison of Language Modeling Approaches for Turkish Broadcast News", *Proceedings of IEEE SIU*, pp. 1 – 4, Didim, Turkey, 2008.

13. Chen, S. F. and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", *Computer Speech and Language*, Vol. 13, No. 4, 1999.

14. Rosenfeld, R., "Two Decades of Statistical Language Modeling: Where Do We Go From Here?", *Proceedings of IEEE*, Vol. 88, pp. 1270 – 1278, 2000.

15. Erguvanlı, E., A. Göksel, and M. Nakipoğlu-Demiralp, "Structure of Modern Turkish", TK 204 class notes, Boğaziçi University, 2003.

16. Oflazer, K. and H. C. Bozşahin, "Turkish Natural Language Processing Initiative: An Overview", *Proceedings of the Third Turkish Symposium on Artificial Intelligence and Artificial Neural Networks*, Ankara, Turkey, 1994.

17. Erguvanlı, E., *The Function of Word Order in Turkish Grammar*, Ph.D. thesis, University of California, Los Angeles, USA, 1979.

18. Sak, H., T. Güngör, and M. Saraçlar, "Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus", *Proceedings of GoTAL, LNAI 5221*, pp. 417 – 427, 2008.

19. Hetherington, I. L., *A Characterization of the Problem of New, Out-of-Vocabulary*

Words in Continuous-Speech Recognition and Understanding, Ph.D. thesis, Massachusetts Institute of Technology, 1995.

20. Rosenfeld, R., "Optimizing Lexical and N-gram Coverage Via Judicious Use of Linguistic Data", *Proceedings of European Conference on Speech Communication and Technology*, pp. 1763 – 1766, 1995.

21. Hirsimäki, T., M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen, "Unlimited Vocabulary Speech Recognition with Morph Language Models Applied to Finnish", *Computer Speech Language*, Vol. 20, No. 4, pp. 515 – 541, 2006.

22. Kurimo, M., A. Puurula, E. Arısoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and M. Saraçlar, "Unlimited Vocabulary Speech Recognition for Agglutinative Languages", *Proceedings of HLT-NAACL*, pp. 487 – 494, New York, USA, 2006.

23. Mihajlik, P., T. Fegyò, Z. Tüske, and P. Ircing, "A Morpho-Graphemic Approach for the Recognition of Spontaneous Speech in Agglutinative Languages - like Hungarian", *Proceedings of Interspeech*, pp. 1497 – 1500, Antwerp, Belgium, August 2007.

24. Podvesky, P. and P. Machek, "Speech Recognition of Czech – Inclusion of Rare Words Helps", *Proceedings of ACL Student Research Workshop*, pp. 121 – 126, Ann Arbor, Michigan, USA, 2005.

25. Hirsimäki, T., J. Pylkkönen, and M. Kurimo, "Importance of High-Order N-Gram Models in Morph-Based Speech Recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 17, No. 4, pp. 724 – 732, 2009.

26. Geutner, P., "Using Morphology Towards Better Large Vocabulary Speech Recognition Systems", *Proceedings of ICASSP*, pp. 445 – 448, Detroit, MI, USA, 1996.

27. Byrne, W., J. Hajic, P. Ircing, F. Jelinek, S. Khudanpur, P. Krbec, and J. Psutka, "On Large Vocabulary Continuous Speech Recognition of Highly Inflectional Language – Czech", *Proceedings of Eurospeech*, pp. 487 – 490, Aalborg, Denmark, 2001.

28. Kwon, O.-W. and J. Park, "Korean Large Vocabulary Continuous Speech Recognition with Morpheme-Based Recognition Units", *Speech Communication*, Vol. 39, pp. 287 – 300, 2003.

29. Kirchhoff, K., D. Vergyri, J. Bilmes, K. Duh, and A. Stolcke, "Morphology-Based Language Modeling for Conversational Arabic Speech Recognition", *Computer Speech and Language*, Vol. 20, No. 4, pp. 589 – 608, 2006.

30. Choueiter, G., D. Povey, S. F. Chen, and G. Zweig, "Morpheme-Based Language Modeling for Arabic", *Proceedings of ICASSP*, Toulouse, France, May 2006.

31. Alumäe, T., *Methods for Estonian Large Vocabulary Speech Recognition*, Ph.D. thesis, Tallinn University of Technology, Tallinn, Estonia, 2006.

32. Kanevsky, D., S. Roukos, and J. Sedivy, "Statistical Language Model for Inflected Languages.", US patent No: 5,835,888, 1998.

33. Rotovnik, T., M. S. Maučec, and Z. Kačic, "Large Vocabulary Continuous Speech Recognition of an Inflected Language Using Stems and Endings", *Speech Communication*, Vol. 49, No. 6, pp. 437 – 452, June 2007.

34. Harris, Z., "Morpheme Boundaries within Words: Report on a Computer Test", *Transformations and Discourse Analysis Papers*, Vol. 73, 1967.

35. Goldsmith, J., "Unsupervised Learning of the Morphology of a Natural Language", *Computational Linguistics*, Vol. 27, No. 2, pp. 153 – 198, 2000.

36. Whittaker, E. and P. Woodland, "Particle-Based Language Modelling", *Proceedings of ICSLP*, Vol. 1, pp. 170 – 173, Beijing, China, October 2000.

37. Creutz, M. and K. Lagus, "Unsupervised Discovery of Morpheme", *Proceedings of ACL*, pp. 21 – 30, 2002.

38. Creutz, M. and K. Lagus, "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 2005.

39. Brent, M. R., "An Efficient, Probabilistically Sound Algorithm for Segmentation and Word Discovery", *Machine Learning*, Vol. 34, pp. 71 – 105, 1999.

40. Pellegrini, T. and L. Lamel, "Using Phonetic Features in Unsupervised Word Decompounding for ASR with Application to a Less-Represented Language", *Proceedings of Interspeech*, pp. 1797 – 1800, Antwerp, Belgium, 2007.

41. Pellegrini, T. and L. Lamel, "Automatic Word Decompounding for ASR in a Morphologically Rich Language: Application to Amharic", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 17, No. 5, 2009.

42. Monson, C., *Paramor: From Paradigm Structure to Natural Language Morphology Induction*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2009.

43. Siivola, V., T. Hirsimäki, M. Creutz, and M. Kurimo, "Unlimited Vocabulary Speech Recognition Based on Morphs Discovered in an Unsupervised Manner", *Proceedings of Eurospeech*, pp. 2293 – 2296, Geneva, Switzerland, 2003.

44. Hirsimäki, T., *Advances in Unlimited Vocabulary Speech Recognition for Morphologically Rich Languages*, Ph.D. thesis, Helsinki University of Technology, Espoo, Finland, 2009.

45. Kneissler, J. and D. Klakow, "Speech Recognition for Huge Vocabularies by Using Optimized Sub-word Units", *Proceedings of Eurospeech*, pp. 69 – 73, Aalborg,

Denmark, 2001.

46. Scharenborg, O., S. Seneff, and L. Boves, "A Two-Pass Approach for Handling Out-of-Vocabulary Words in a Large Vocabulary Recognition Task", *Computer Speech Language*, Vol. 21, No. 1, pp. 206 – 218, 2007.

47. Scharenborg, O., D. Norris, L. ten Bosch, and J. M. McQueenc, "How Should a Speech Recognizer Work?", *Cognitive Science*, Vol. 29, pp. 867 – 918, 2005.

48. Hetherington, I. L., "A Multi-Pass, Dynamic-Vocabulary Approach to Real-Time, Large-Vocabulary Speech Recognition", *Proceedings of Interspeech*, pp. 545 – 548, Lisbon, Portugal, 2005.

49. Yazgan, A. and M. Saraçlar, "Hybrid Language Models for Out of Vocabulary Word Detection in Large Vocabulary Conversational Speech Recognition", *Proceedings of ICASSP*, Montreal, Canada, 2004.

50. Ircing, P. and J. Psutka, "Two-Pass Recognition of Czech Speech Using Adaptive Vocabulary", *Proceedings of TSD*, pp. 273 – 277, Czech Republic, 2001.

51. Chen, L., J.-L. Gauvain, L. Lamel, and G. Adda, "Unsupervised Language Model Adaptation for Broadcast News", *Proceedings of ICASSP*, pp. 220 – 223, 2003.

52. Martins, C., A. Texeira, and J. Neto, "Dynamic Vocabulary Adaptation for a Daily and Real-Time Broadcast News Transcription System", *IEEE Spoken Language Technology Workshop*, pp. 146 – 149, Palm Beach, Aruba, 2006.

53. Ohtsuki, K., N. Hiroshima, M. Oku, and A. Imamura, "Unsupervised Vocabulary Expansion for Automatic Transcription of Broadcast News", *Proceedings of ICASSP*, pp. 1021 – 1024, Philadelphia, PA, USA, 2005.

54. Palmer, D. and M. Ostendorf, "Improving Out-of-Vocabulary Name Resolution", *Computer Speech Language*, Vol. 19, No. 1, pp. 107 – 128, 2005.

55. Geutner, P., M. Finke, P. Scheytt, A. Waibel, and H. Wactlar, "Transcribing Multilingual Broadcast News Using Hypothesis Driven Lexical Adaptation", *Proceedings of DARPA Broadcast News Workshop*, Herndon, USA, 1998.

56. Geutner, P., M. Finke, and A. Waibel, "Phonetic-Distance-Based Hypothesis Driven Lexical Adaptation for Transcribing Multilingual Broadcast News", *Proceedings of ICSLP*, Sydney, Australia, 1998.

57. Geutner, P., M. Finke, and A. Waibel, "Selection Criteria for Hypothesis Driven Lexical Adaptation", *Proceedings of ICASSP*, Phoenix, Arizona, USA, 1999.

58. Brown, P. F., V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-Based N-Gram Models of Natural Language", *Computational Linguistics*, Vol. 18, No. 4, 1990.

59. Chelba, C. and F. Jelinek, "Structured Language Modeling", *Computer Speech and Language*, Vol. 14, No. 4, pp. 283 – 332, 2000.

60. Roark, B., "Probabilistic Top-Down Parsing and Language Modeling", *Computational Linguistics*, Vol. 27, No. 2, pp. 249 – 276, 2001.

61. Wang, W. and M. P. Harper, "The SuperARV Language Model: Investigating the Effectiveness of Tightly Integrating Multiple Knowledge Sources", *Proceedings of EMNLP*, pp. 238 – 247, Philadelphia, PA, USA, 2002.

62. Bilmes, J. A. and K. Kirchhoff, "Factored Language Models and Generalized Parallel Backoff", *Proceedings of NAACL*, pp. 4 – 6, Edmonton, Canada, 2003.

63. Rosenfeld, R., *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

64. Khudanpur, S. and J. Wu, "Maximum Entropy Techniques for Exploiting Syntactic, Semantic and Collocational Dependencies in Language Modeling", *Computer Speech Language*, Vol. 14, pp. 355 – 372, 2000.

65. Erdoğan, H., R. Sarıkaya, S. F. Chen, Y. Gao, and M. Picheny, "Using Semantic Analysis to Improve Speech Recognition Performance", *Computer Speech Language*, Vol. 19, pp. 321 – 343, 2005.

66. Sarıkaya, R., M. Afify, D. Yonggang, H. Erdoğan, and G. Yuqing, "Joint Morphological-Lexical Language Modeling for Processing Morphologically Rich Languages With Application to Dialectal Arabic", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 7, pp. 1330 – 1339, 2008.

67. Rosenfeld, R., S. F. Chen, and X. Zhu, "Whole-Sentence Exponential Language Models: a Vehicle for Linguistic-Statistical Integration", *Computer Speech and Language*, Vol. 15, No. 1, 2001.

68. Roark, B., M. Saraçlar, and M. Collins, "Discriminative N-gram Language Modeling", *Computer Speech and Language*, Vol. 21, No. 2, pp. 373 – 392, 2007.

69. Roark, B., M. Saraçlar, M. J. Collins, and M. Johnson, "Discriminative Language Modeling with Conditional Random Fields and the Perceptron Algorithm", *Proceedings of ACL*, pp. 47 – 54, Barcelona, Spain, 2004.

70. Collins, M., M. Saraçlar, and B. Roark, "Discriminative Syntactic Language Modeling for Speech Recognition", *Proceedings of ACL*, pp. 507 – 514, Ann Arbor, MI, USA, 2005.

71. Shafran, I. and K. Hall, "Corrective Models for Speech Recognition of Inflected Languages", *Proceedings of EMNLP*, pp. 390 – 398, Sydney, Australia, 2006.

72. Singh-Miller, N. and M. Collins, "Trigger-Based Language Modeling Using a Loss-Sensitive Perceptron Algorithm", *Proceedings of ICASSP*, pp. 25 – 28, Honolulu, Hawaii, USA, 2007.

73. Povey, D. and P. C. Woodland, "Large-Scale MMIE Training for Conversational Telephone Speech Recognition", *Proceedings of NIST Speech Transcription Work-*

shop, College Park, MD, USA, 2000.

74. Povey, D. and P. C. Woodland, "Minimum Phone Error and I-Smoothing for Improved Discriminative Training", *Proceedings of ICASSP*, pp. 105 – 108, Orlando, FL, USA, 2002.

75. Lin, S.-S. and F. Yvon, "Discriminative Training of Finite State Decoding Graphs", *Proceedings of Interspeech*, pp. 733 – 736, Lisbon, Portugal, 2005.

76. Kuo, H.-K., B. Kingsbury, and G. Zweig, "Discriminative Training of Decoding Graphs for Large Vocabulary Continuous Speech Recognition", *Proceedings of ICASSP*, Vol. 4, pp. 45 – 48, Honolulu, Hawaii, USA, April 2007.

77. Çarkı, K., P. Geutner, and T. Schultz, "Turkish LVCSR: Towards Better Speech Recognition for Agglutinative Languages", *Proceedings of ICASSP*, pp. 1563 – 1566, İstanbul, Turkey, 2000.

78. Hakkani-Tür, D. Z., *Statistical Language Modeling for Agglutinative Languages*, Ph.D. thesis, Bilkent University, Ankara, Turkey, 2000.

79. Mengüşoğlu, E. and O. Deroo, "Turkish LVCSR: Database Preparation and Language Modeling for an Agglutinative Language", *Proceedings of ICASSP, Student Forum*, Salt-Lake City, UT, USA, 2001.

80. Dutağacı, H., *Statistical Language Models for Large Vocabulary Continuous Speech Recognition of Turkish*, Master's thesis, Boğaziçi University, İstanbul, Turkey, 2002.

81. Arısoy, E., *Turkish Dictation System for Radiology and Broadcast News Applications*, Master's thesis, Boğaziçi University, İstanbul, Turkey, 2004.

82. Arısoy, E., H. Dutağacı, and L. M. Arslan, "A Unified Language Model for Large Vocabulary Continuous Speech Recognition of Turkish", *Signal Processing*, Vol. 86, No. 10, pp. 2844 – 2862, 2006.

83. Çiloğlu, T., M. Çömez, and S. Şahin, "Language Modeling for Turkish as an Agglutinative Language", *Proceedings of IEEE SIU*, pp. 461 – 462, Kuşadası, Turkey, 2004.

84. Bayer, A. O., T. Çiloğlu, and M. T. Yöndem, "Investigation of Different Language Models for Turkish Speech Recognition", *Proceedings of IEEE SIU*, pp. 1 – 4, Antalya, Turkey, 2006.

85. Hacıoğlu, K., B. Pellom, T. Çiloğlu, Ö. Öztürk, M. Kurimo, and M. Creutz, "On Lexicon Creation for Turkish LVCSR", *Proceedings of Eurospeech*, pp. 1165 – 1168, Geneva, Switzerland, 2003.

86. Duh, K. and K. Kirchhoff, "Automatic Learning of Language Model Structure", *Proceedings of COLING*, Geneva, Switzerland, 2004.

87. Erdoğan, H., O. Büyük, and K. Oflazer, "Incorporating Language Constraints in Sub-Word Based Speech Recognition", *Proceedings of ASRU*, pp. 98 – 103, San Juan, Puerto Rico, 2005.

88. Strassel, S., D. Miller, K. Walker, and C. Cieri, "Shared Resources for Robust Speech-to-Text Technology", *Proceedings of Eurospeech*, pp. 1609 – 1612, Geneva, Switzerland, September 2003.

89. Salor, Ö., B. L. Pellom, T. Çiloğlu, and M. Demirekler, "Turkish Speech Corpora and Recognition Tools Developed by Porting SONIC: Towards Multilingual Speech Recognition", *Computer Speech Language*, Vol. 21, No. 4, pp. 580 – 593, 2007.

90. Fromkin, V., R. Rodman, and N. Hyams, *An Introduction to Language*, Thomson Heinle, Massachusetts, 2003.

91. Çetinoğlu, Ö., *Prolog Based Natural Language Processing Infrastructure for Turkish*, Master's thesis, Boğaziçi University, İstanbul, Turkey, 2000.

92. Oflazer, K., "Two-level Description of Turkish Morphology", *Literary and Linguistic Computing*, Vol. 9, No. 2, pp. 137 – 148, 1994.

93. Hakkani-Tür, D., K. Oflazer, and G. Tür, "Statistical Morphological Disambiguation for Agglutinative Languages", *Journal of Computers and Humanities*, Vol. 36, No. 4, pp. 381 – 410, 2002.

94. Yüret, D. and F. Türe, "Learning Morphological Disambiguation Rules for Turkish", *Proceedings of HLT-NAACL*, pp. 328 – 334, New York, USA, 2006.

95. Sak, H., T.Güngör, and M. Saraçlar, "Morphological Disambiguation of Turkish Text with Perceptron Algorithm", *CICLing, LNCS 4394*, pp. 107 – 118, 2007.

96. Eryiğit, G., J. Nivre, and K. Oflazer, "Dependency Parsing of Turkish", *Computational Linguistics*, Vol. 34, No. 3, pp. 357 – 389, 2008.

97. Young, S., D. Ollason, V. Valtchev, and P. Woodland, "The HTK Book (for HTK Version 3.2), Entropic Cambridge Research Laboratory", 2002.

98. Mohri, M., F. C. Pereira, and M. D. Riley, "AT&T FSM Libary, Finite State Machine Library, AT&T Labs Research. http://www.research.att.com/sw/tools/fsm/", 2002.

99. Stolcke, A., "SRILM – An Extensible Language Modeling Toolkit", *Proceedings of ICSLP*, Vol. 2, pp. 901 – 904, Denver, 2002.

100. Stolcke, A., "Entropy-Based Pruning of Backoff Language Models", *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pp. 270 – 274, Lansdowne, VA, USA, 1998.

101. Mohri, M. and M. D. Riley, "DCD Library, Speech Recognition Decoder Library, AT&T Labs – Research. http://www.research.att.com/sw/tools/dcd/", 2002.

102. Guz, U., B. Favre, D. Hakkani-Tür, and G. Tur, "Generative and Discriminative

Methods Using Morphological Information for Sentence Segmentation of Turkish", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 17, No. 5, pp. 895 – 903, 2009.

103. Creutz, M., T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arısoy, M. Saraçlar, and A. Stolcke, "Analysis of Morph-Based Speech Recognition and the Modeling of Out-of-Vocabulary Words Across Languages", *Proceedings of HLT-NAACL*, pp. 380 – 387, Rochester, NY, USA, 2007.

104. Creutz, M., T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arısoy, M. Saraçlar, and A. Stolcke, "Morph-Based Speech Recognition and Modeling of Out-of-Vocabulary Words Across Languages", *ACM Transactions on Speech and Language Processing*, Vol. 5, No. 1, pp. 1 – 29, 2007.

105. Arısoy, E., T. Pellegrini, M. Sraçlar, and L. Lamel, "Enhanced Morfessor Algorithm with Phonetic Features: Application to Turkish", *Proceedings of SPECOM*, St. Petersburg, Russia, 2009.

106. Sak, H., M. Saraçlar, and T. Güngör, "Morphology-Based and Sub-Word Language Modeling for Turkish Speech Recognition", *Proceedings of ICASSP*, Dallas, Texas, USA, 2010.

107. Pallett, D. S., W. M. Fisher, and J. G. Fiscus, "Tools for the Analysis of Benchmark Speech Recognition Tests", *Proceedings of ICASSP*, pp. 97 – 100, Albuquerque, New Mexico, USA, 1990.

108. Chase, L., *Error-Responsive Feedback Mechanisms for Speech Recognizers*, Ph.D. thesis, Carnegie-Mellon University, Pittsburgh, USA, 1997.

109. Hirsimäki, T. and M. Kurimo, "Analysing Recognition Errors in Unlimited-Vocabulary Speech Recognition", *Proceedings of HLT-NAACL*, pp. 193 – 196, Boulder, Colorado, USA, June 2009.

110. Bahl, L., P. Brown, P. deSouza, and R. Mercer, "Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition", *Proceedings of ICASSP*, pp. 49 – 52, Tokyo, Japan, 1986.

111. Lafferty, J. D., A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *Proceedings of ICML*, pp. 282 – 289, Williams, MA, USA, 2001.

112. Berger, A. L., S. D. Della Pietra, and V. J. D. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing", *Computational Linguistics*, Vol. 22, No. 1, pp. 39 – 71, 1996.

113. Collins, M., "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms", *Proceedings of EMNLP*, pp. 1 – 8, Philadelphia, PA, USA, 2002.

114. Freud, Y. and R. E. Schapire, "Large Margin Classification using the Perceptron Algorithm", *Machine Learning*, Vol. 37, No. 3, pp. 277 – 296, 1999.

115. Sarikaya, R., M. Afify, Y. Deng, H. Erdogan, and Y. Gao, "Joint Morphological-Lexical Modeling for Processing Morphologically-Rich Languages", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 7, 2008.

116. Sak, H., T. Gungor, and M. Saraclar, "Morphological Disambiguation of Turkish Text with Perceptron Algorithm", *Proceedings of CICLing*, Mexico City, Mexico, 2007.

117. Manning, C. and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA, 1999.

118. Jelinek, F., B. Merialdo, S. Roukos, and M. Strauss, "A Dynamic Language Model for Speech Recognition", *Proceedings of HLT, the Workshop on Speech and Natural Language*, pp. 293 – 295, Pacific Grove, California, USA, 1991.

119. Bacchiani, M., B. Roark, and M. Saraçlar, "Language Model Adaptation with MAP Estimation and the Perceptron Algorithm", *Proceedings of HLT-NAACL*, Boston, MA, USA, 2004.