Automatic Speech Recognition of Agglutinative Language based on Lexicon Optimization

Mijiti Abulimiti

Automatic Speech Recognition of Agglutinative Language based on Lexicon Optimization

Mijit Ablimit

Graduate School of Informatics

Kyoto University

Kyoto, Japan

Abstract

In *agglutinative* languages, selection of lexical unit is not obvious and it is one of the important issues in designing a language model for automatic speech recognition (ASR). Choice of lexicon unit affects unit length, frequency and also ASR performance. Words in *agglutinative* languages have a variety of derivatives, and increase the vocabulary size explosively, causing OOV (out-of-vocabulary) and data sparseness problems. Therefore, the morpheme unit is conventionally adopted in many inflectional languages, such as Japanese, Korean, Turkish, Finnish, German and Arabic. However, morphemes are short, often consisting of one or two phonemes, thus they are more likely to be confused in ASR than the word unit. The goal of this study is to design an optimal lexicon by balancing the characteristics of the word unit and the morpheme unit.

As opposed to the previous work, we investigate approaches that are directly related to ASR performance or WER (word error rate), considering phonetic confusability and unit length. A novel discriminative approach is proposed to select word or sub-word entries which are likely to reduce the WER. We compare segmentation and concatenation approaches for lexicon optimization in our Uyghur large-vocabulary continuous speech recognition (LVCSR) system. The proposed concatenation method based on the discriminative approach significantly reduces WER and the lexicon size compared to the word-based model. Furthermore, the method preserves linguistic information including both word and morpheme boundaries, which is useful for many natural language applications such as machine translation and information retrieval.

Chapter 1 provides an overview of the problems in lexicon design in *agglutinative* languages and approaches investigated in the thesis.

Chapter 2 briefly reviews previous studies on lexicon optimization. They are based on statistical measures, such as co-occurrence frequency, mutual probability and language model likelihood. Unsupervised morpheme extraction methods are also reviewed.

Chapter 3 presents our morphological analyzer based on rules and statistical models. The morpheme segmentation accuracy of 97.6% is achieved and a baseline ASR system is built based on the morpheme and word units. Statistical characteristics and baseline ASR performance of various unit sets are compared.

Based on the baseline ASR results, Chapter 4 presents an approach which extracts problematic morpheme sequences. It is realized by aligning and comparing the ASR results by the morpheme-based model with those by the word-based model. The manually selected features reduce both WER and the lexicon size while maintaining the high coverage of the morpheme unit. The combined effect of these features is also confirmed.

In Chapter 5, we present a discriminative approach to automate the feature extraction from the two layers of ASR results. We describe each word by a set of features and define an evaluation function. Then, the weights of the features are learned by the difference of the two ASR results to select "critical" word entries which generate different (probably correct) hypotheses from the morpheme-based unit. This learning mechanism is applicable to any unseen words or sub-words. Specifically, we apply the Support Vector Machines (SVM) and Logistic Regression (LR) model as well as simple perceptron. The SVM and LR are more robustly trained and SVM results in the best performance, significantly reducing both WER and the lexicon size. The proposed learning scheme is realized in an unsupervised manner in that it does not need correct transcription for training data.

Chapter 6 compares various sub-word optimization approaches in the Uyghur LVCSR system. Both segmentation and concatenation methods with manual and automatic approaches are compared. Although the best performances of both methods are comparable, the proposed morpheme concatenation approach is advantageous in that it preserves linguistic information.

Chapter 7 concludes the thesis with discussions of the generality of the proposed approach in other *agglutinative* languages.

Acknowledgements

First and foremost, I sincerely express my thanks to my supervisor Professor Tatsuya Kawahara. When I start my PhD. study at Kyoto University, I was incompetent for the new standard of research level. Under his patient guidance I gradually learned much invaluable knowledge covering from the small details to overall organization, such as how to set up experiments, how to present results, and how to write an excellent academic paper. For the first time, we have successfully built an LVCSR system for Uyghur language from a scratch, which is the most challenging task in terms of the scarcity of resource and quality of the corpora, and this work took a lot of our time. With Kawahara sensei's strong confident support and guidance, we are able to accomplish high-level academic works from a very poor starting condition, and able to publish top-quality technical papers with innovative proposals.

I am deeply grateful to Professor Shinsuke Mori and Dr. Graham Neubig for their generous help. They have given me many invaluable suggestions, and taught me many things that are essential to my research works. I am very lucky for being in the same laboratory with them, and able to share their rich academic knowledge and the tools built by them.

I would like to thank to Professor Askar Hamdulla, for his enormous work of collecting and preparing all the corpora which are necessary for us. I sincerely thank him for his unwavering support throughout all my research works and all the friendship he has offered through the years.

I also owe a great deal to the members of Kawahara laboratory, thank you for all your help, and the great times we have spent together.

Thanks to my relatives and my friends, they have been supportive for my study in Japan. And thanks to many others who have been supportive of the research program.

Contents

List of Figures

List of Tables

List of Symbols and Acronyms

Symbols

t_i	i^{th} unit in a sequence of units
t_{i-n+1}^{i-1}	Previous n-1 units of t_i
$H(S)$	Entropy of text sequence S
PP	Perplexity
PP^*	Normalized perplexity
Φ_{freq}	Feature based on unit frequency
Φ_{length}	Feature based on unit length
Φ_{stem}	Feature based on stem
Φ_{word_ending}	Feature based on word ending
$\Phi_{\text{bigram\_}m_i\,m_j}$	Feature based on bigram $m_i\,m_j$
$\Phi_s(w)$	An individual feature of word unit w
$\Phi(w)$	Feature set of word unit w
$(\Phi(w^i), y^i)$	Feature and desired value of i^{th} sample
α_s	Weight of feature $\Phi_s(w^i)$
α	Weight vector
$\xi(-)$	Loss function
α^T	Transpose of vector α
Cutoff-F	Units whose frequency is less than F are considered unknown
m_i	A morpheme unit in a sequence of morphemes
st_i	A stem sample
we_i	A word-ending sample
$C(t_i\,t_j)$	Co-occurrence frequency of $t_i\,t_j$
statistical morpheme	Morpheme like units extracted in an unsupervised way from a raw text corpus
Morfessor	Unsupervised morpheme segmentation tool developed by Creutz et al.

Acronyms

AM	Acoustic model
ASR	Automatic speech recognition
CER	Character error rate
DLM	Discriminative language model
FMS	Frequent morpheme sequence
GCLM	Global conditional log-linear model
GPD	Generalized probabilistic descent
HMM	Hidden Markov model
LM	Language model
LR	Logistic regression
LVCSR	Large vocabulary continuous speech recognition
MAP	Maximum a posteriori
MB	Mutual bigram
MCE	Minimum classification error
MDL	Minimum description length
MER	Morpheme error rate
MFCC	Mel-frequency Cepstral coefficient
MI	Mutual information
ML	Maximum likelihood
MMIE	Maximum Mutual Information Estimation
MP	Mutual probability
MT	Machine translation
NLP	Natural language processing
OOV	Out of Vocabulary
SER	Syllable error rate
SVM	Support vector machines
WER	Word error rate

Chapter 1

Introduction

1.1 Background

Automatic speech recognition (ASR) systems based on statistical paradigm have been a great success. It is already a common practice for people to be talking to many kinds of devices, applications and services as they move around their homes, offices and cities, which are only a vision a few decades ago. Speech interfaces can realize natural, intuitive and flexible interaction, and provide a basis for intelligent systems. The goal of ASR is to transcribe spoken utterances into written text or sentences. Large-vocabulary continuous speech recognition (LVCSR) systems can produce high-quality transcriptions for wide domains, and have already been applied to many different languages.

However, there are still a number of languages for which LVCSR systems have not been developed. One of the critical problems is limited data availability for training statistical models. Another problem is that the word unit is not universally defined in all languages. Especially, words in *agglutinative* languages are relatively long, and the vocabulary size of these languages is growing up exponentially. In some languages like Japanese, there is no clear word delimiter. These problems pose a challenge in applying word units-based modeling to *agglutinative* languages together with the data sparseness problem.

1.2 Problems of lexicon design in agglutinative languages

In *agglutinative* languages, sub-word units such as morphemes or morpheme-like units are adopted to alleviate the data sparseness and to reduce out-of-vocabulary (OOV) problems [16][72]. Morphemes are selected as the appropriate unit for highly inflectional languages, as they can provide better coverage and a smaller vocabulary size. Furthermore, morphemes provide linguistic information, including the word and morpheme boundaries, which is important for adding meaningful constraint to the statistical framework and convenient for downstream natural language processing

(NLP). However, formulation with short units shrinks the context of the sequential probabilistic models, and may cause degradation in ASR [14]. Even smaller units like syllables are too short, easily confused, and also difficult to carry linguistic information.

Modeling based on long and short units have their merits and demerits respectively, which can be explained by both statistical and linguistic aspects [6][55]. Therefore, analyzing these reasons and finding an optimal unit set, which has both high coverage and better constraint, is a very important research topic for highly inflectional languages. The statistical models also encounter the problem of insufficient training data. The proper unit set may increase the reliability of the statistical estimation.

Sub-word based approaches are reported to have improved ASR performance [6][55]. The main idea of these methods is concatenating frequently co-occurred units to enhance linguistic constraint, while segmenting less frequent and long units to reduce OOV rate. However, the unit selection based on only frequency may lead to over-generalization, and does not consider linguistic constraints. Furthermore, these data-driven approaches based on simple parameters like co-occurrence frequency and length do not always outperform the naive word based ASR system, especially when a larger training corpus is available [14][87].

Some unsupervised morpheme segmentation methods are developed to avoid the manual labor needed for the annotated corpus and the phonetic and morphological analysis. *Statistical morphemes* (or pseudo morphemes) are extracted from a raw text corpus in an unsupervised way by using probabilistic methods such as maximum a posteriori (MAP) [55]. Totally based on a raw text corpus, the joint probability of the optimized sub-word unit sequence is maximized in an unsupervised manner. This method is applied to many inflectional languages and reported to have improved ASR performance [58][59]. But the *statistical morpheme*s are actually random units without considering linguistic information, and the selected unit and the vocabulary size are dependent on the training corpus [57].

Choosing an optimal lexicon is not a straight-forward splitting of long units, because there are some phonological and morphological changes. And the linguistic information such as the word boundary should be preserved. One direct solution is to keep the co-articulated and problematic boundaries inside the units when we select the lexical units. The long units like phrases or words suffer less co-articulation problems than the short units like morphemes. In the meantime, we must prevent from over-concatenation which easily leads to vocabulary explosion.

As the ASR system is a complicated probabilistic framework, and which involves phonetic matching and language model, its performance is affected by the phonetic similarity and the linguistic constrains. In order to define an optimal lexicon which can produce the best results (smallest WER and lexicon size in this thesis), we need to consider a variety of factors affecting ASR. However, most of the previous concatenation or segmentation approaches are purely based on statistical characteristics of text, and do not consider phonetic confusability or morphological changes and co-articulation effects [6]. This is the fundamental problem in the existing approaches.

1.3 Approaches to lexicon optimization

Based on the above consideration, in this thesis, we adopt the following approaches.

1.3.1 Linguistic morpheme unit segmentation

Because of the explosive nature in the word unit of the *agglutinative* languages, using sub-word linguistic unit is essential. The *agglutinative* languages have several layers of linguistic sub-word units such as morpheme, syllable and phoneme. We adopt *linguistic morphemes*, because they are a meaningful unit for NSP applications such as information retrieval (IR) and machine translation (MT). As there are no delimiters for the morpheme unit, rule-based or statistical methods are adopted to segment words into the morphological unit. A supervised segmentation method is necessary for *linguistic morpheme* segmentation, and it is trained with a manually prepared training corpus.

1.3.2 Feature extraction from two layers of ASR results

The morpheme sequences which are easily misrecognized by the ASR system can be enhanced by merging into longer units. In contrast to previous approaches purely based on statistics of text, we propose a novel approach which directly considers ASR performance. This can be done by comparing ASR results of two different basic unit sets of word and morpheme. We transcribe a large speech corpus separately into two texts based on words and morphemes. These two results are then aligned in order to extract problematic morpheme sequences. Here we assume the word unit provides better ASR performance.

The problematic morpheme sequences can be extracted based on several features such as frequency, length, n-gram sequences. This method can take into consideration both the phonetic confusion and linguistic constraint. Therefore, it is related with the ASR performance, and can incorporate effective word or sub-word entries selectively while maintaining the high coverage of the morpheme unit.

1.3.3 Discriminative approach to morpheme concatenation

We automate the process of extracting and modeling of the problematic morphological sequences. We propose a novel discriminative approach. It is based on an evaluation function for each word defined by a set of features and their weights, which are optimized by the difference of word error rates (WERs) by the morpheme-based model and those by the word-based model. Then, word (or sub-word) entries with higher evaluation scores are selected to be added to the lexicon. We investigate several discriminative models to realize this scheme. Specifically, we implement it with Support Vector Machines (SVM) and Logistic Regression (LR) model as well as simple perceptron.

A large training speech corpus is necessary for this method, and the speech corpus used for acoustic modeling can be re-used. Furthermore, this learning scheme can also be realized in an unsupervised manner in that it does not need correct transcription for training data.

Figure 1 shows an overall framework of lexicon optimization addressed in this thesis. First a morphological analyzer is trained to build ASR systems based on the morpheme and word units. Then, the feature extraction and discriminative learning are applied. These processes concatenate the morpheme sequences into optimal sub-word units of the final lexicon.

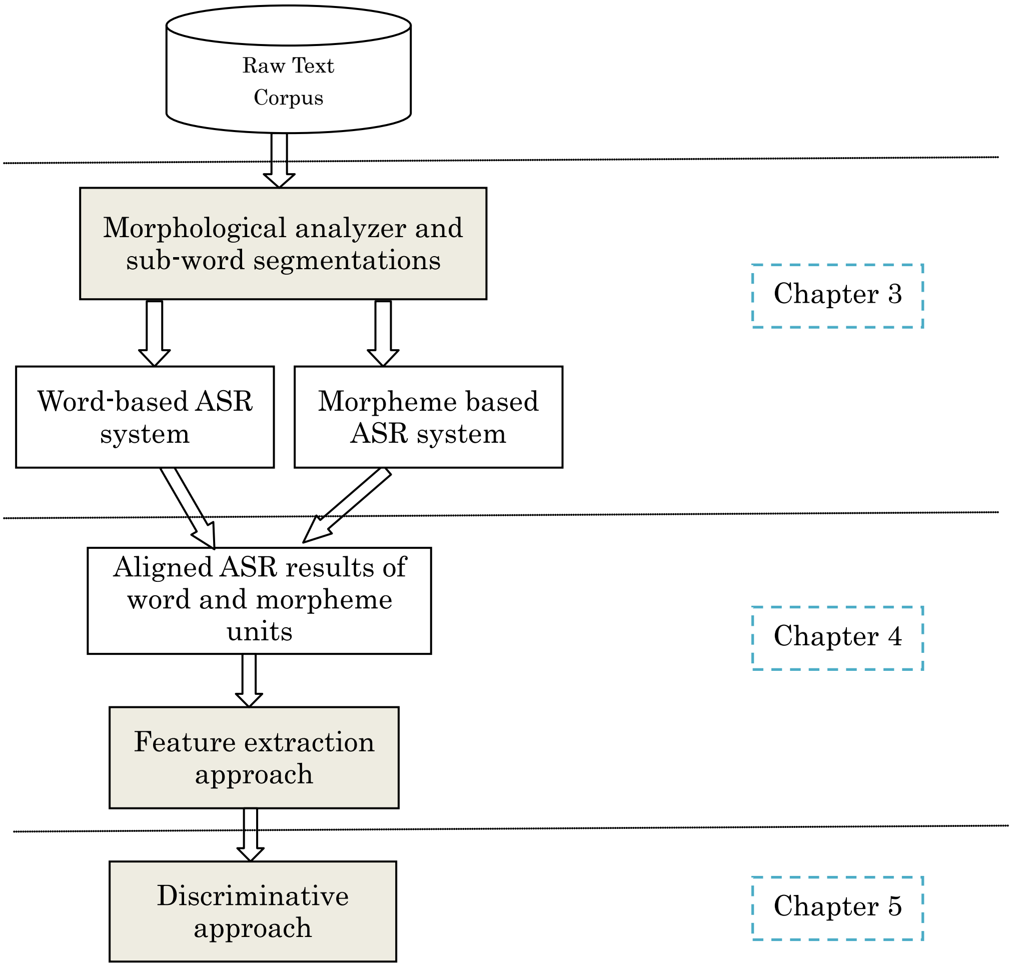

Raw Text
Corpus
Morphological analyzer and sub-word segmentations
Chapter 3
Word-based ASR system
Morpheme based ASR system
Aligned ASR results of word and morpheme units
Chapter 4
Feature extraction approach
Discriminative approach
Chapter 5

Figure 1.1 Flow chart of the proposed lexicon optimization.

Chapter 2

Review on lexicon optimization methods

There are many studies that explore appropriate units to improve the ASR performance. The majority of them are data driven methods based on some statistics such as frequency, mutual information and perplexity. There are also studies on unsupervised methods for selecting basic units. These are reviewed in this Chapter.

2.1 Overview of automatic speech recognition (ASR)

ASR is formulized as a task of finding a series of linguistic units such as sentences (or texts) S given the acoustic features U of a speech signal containing these sentences. Statistical ASR systems create a model to compute the posterior probability of the sentences S given the acoustic features U, and searching for the sequence that maximizes this probability.

$$\hat{S} = \mathrm{argmax}_S \, P(S|U) \tag{2-1}$$

As the posterior probability is difficult to model directly, Bayes's law is used to decompose the probability

$$\hat{S} = \mathrm{argmax}_S \frac{P(U|S)P(S)}{P(U)} = \mathrm{argmax}_S \, P(U|S)P(S) \tag{2-2}$$

Here, $P(U|S)$ is the probabilistic connection between sentences and their acoustic features. The dimensionality of various sentences is actually infinite, so it is unrealistic to directly modeling all the different sentences due to data sparseness. Instead, acoustic model (AM) is trained to recognize sequence of phonemes X, which can then be mapped into the text (or sentence) S. Phonemes are defined as the smallest perceptible linguistic unit of speech. Thus, the entire ASR process is usually further approximated by choosing the single most likely phoneme sequence to allow for efficient search.

$$\hat{S} \approx \mathrm{argmax}_{S,X} \, P(U|X)P(X|S)P(S) \tag{2-3}$$

Here, $P(U|X)$ indicates the AM probability and $P(X|S)$ is a lexical model that maps between sentences and their pronunciations. $P(S)$ is computed with a language model (LM). It should be noted that a weight factor α is used for adjustment as $P(S)^{\alpha}$. A larger weight indicates the LM will have a larger influence on the recognition results than the AM, and vice versa.

Large training text and speech data is necessary to build language model and acoustic model, as shown in Figure 2.1. And a decoding program will transcribe the speech into text using these models.

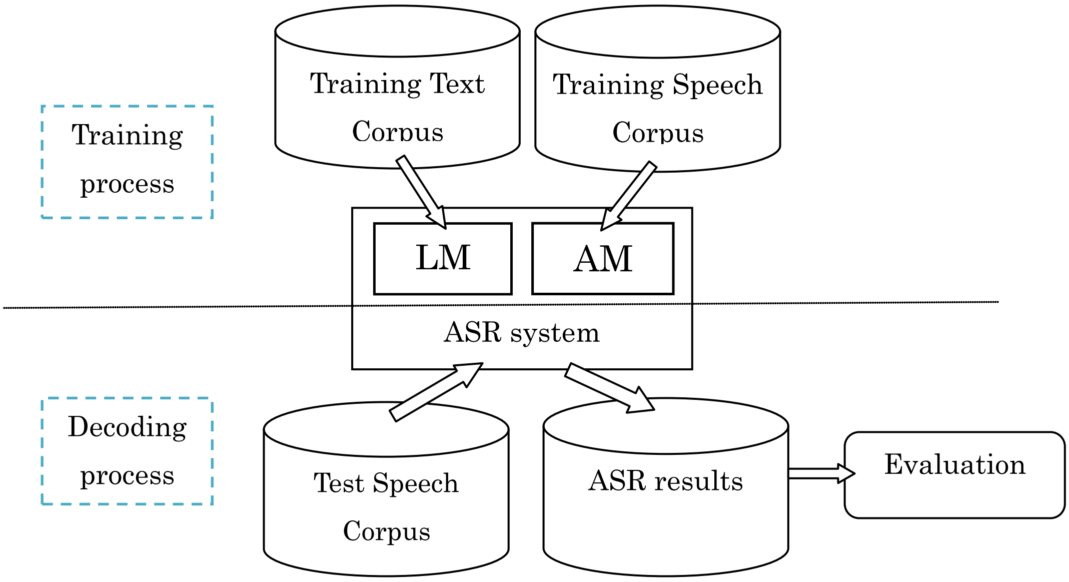


Figure 2.1 Pipeline of ASR system and necessary resources.

2.1.1 Acoustic modeling

The goal of the acoustic model is to estimate the probability $P(U|X)$ for any acoustic evidence U and hypothesized phoneme string X. The acoustic evidence U consists of a sequence of feature vectors extracted from the speech signal. Mel Frequency Cepstral Coefficients (MFCC) feature is the mostly used as an acoustic feature in current ASR systems. Generally, acoustic models are based on the Hidden Markov Model (HMM). An overview of Hidden Markov Models is given in Jelinek [110].

2.1.2 Lexical modeling

Knowledge about the adopted set of lexical units and their pronunciations are represented through the lexical model, or simply the lexicon. The lexicon is a dictionary of all the lexical units with their corresponding pronunciations in the vocabulary of the ASR system. The lexicon consists of one or more basic pronunciations for each unit. The alternative pronunciation can be created by applying phonological rules to the basic forms. The phonological rules can be assimilation (disharmony), harmony etc. Various pronunciations can be formulated with different weights. But in this research, no pronunciation weights are adopted, for the phonological and morphological changes are explicitly considered in our morphological analyzer, and in the lexicon optimization.

2.1.3 Language modeling

The language model (LM) is an essential part of ASR systems, providing linguistic constraints to the recognizer and helping to resolve the ambiguity inherent in the acoustic signal. Widely-used LM is a probabilistic framework of linguistic units. The probability of a sentence $P(S)$ can be expressed as the joint probability of the linguistic units.

$$P(S) = P(t_1 t_2 \dots t_N) = \prod_{i=1}^{N} P(t_i | t_1 \dots t_{i-1}) \qquad (2\text{-}4)$$

Here, the joint probability of sequential units $P(t_1 t_2 \dots t_N)$ is calculated with the product rule conditioned on the previous observations $P(t_i | t_1 \dots t_{i-1})$. Still, the dimension of previous observations can be very large that it is impractical to cover them all. A way to simplify this problem is to assume that a unit t_i is independent of units far from it. Thus, the previous formulation can be approximated by the multiplication of conditional distributions in which each unit is only dependent on the previous $(n-1)$ units.

$$P(t_1 t_2 \dots t_N) \approx \prod_{i=1}^{N} P(t_i | t_{i-n+1}^{i-1}) \qquad (2\text{-}5)$$

This formulation is called n-gram LM which is conditioned only on previous $(n-1)$ observations t_{i-n+1}^{i-1} . Yet, the dimension of n-gram cannot be too large, for it leads to data sparseness issue. Practically, up until 3-gram or 4-gram is used in LM for ASR.

This probability $P(t_i|t_{i-n+1}^{i-1})$ can be easily obtained by counting all the n-grams C(t_{i-n+1}^{i}) and (n-1)-grams C(t_{i-n+1}^{i-1}) in the LM training text data.

$$P\left(t_i\middle|t_{i-n+1}^{i-1}\right) = \frac{C(t_{i-n+1}^{i-1})}{C(t_{i-n+1}^{i})} \tag{2-6}$$

The sequence which represents the text S can be long units like words or phrases, or the short units like syllable and morphemes.

$$P(S) = P(t_1 t_2 \dots t_N) = P(r_1 r_2 \dots r_K) \tag{2-7}$$

where ***t*** is a longer unit set than ***r*** , $N < K$

With the limited training corpus size, shorter unit sequence r_{i-n+1}^{i} provides better coverage, while the longer sequence t_{i-n+1}^{i} catches longer context. There is a trade-off between long and short units.

To avoid assigning zero probabilities for unseen n-grams C(t_{i-n+1}^{i}), it is necessary to smooth the n-gram frequencies. Many approaches have been proposed, additive smoothing [109], linear interpolation [110], Katz Smoothing [112], Witten-Bell Smoothing [113], Absolute Discounting, and Kneser-Ney smoothing modified Kneser-Ney smoothing [5] and so on. The Kneser-Ney smoothing has been shown to outperform all other approaches, and is the most commonly used technique. For more detailed overview of these techniques, refer to Chen and Goodman [114]. Smoothing alleviates the data sparseness problem for unseen n-gram histories. The probability of predicting a novel word (OOV) is obtained by assigning all rare words in the training text to a special <UNK> symbol which is included in the vocabulary. Other approaches include explicitly modeling the OOVs as discussed in [59][66]. The probability for <UNK> needs to be adjusted by dividing by the expected number of unseen words, especially when we compare LMs of different vocabulary sizes.

Choosing the lexical unit t is an important first step, especially for the highly inflectional languages. Word units are relatively long in the inflectional languages, easily cause OOV problem. Choosing smaller units can reduce the OOV problem, but may cause confusion for ASR applications.

2.1.4 Evaluations of ASR systems

There are several measures for evaluating performance of ASR systems. Word error rate (WER) is the direct measure for ASR systems. Out of vocabulary (OOV) rate and perplexity are indirect measures of statistical modeling. The lexicon size is also practically important.

As the word is not the only optional unit, there are different unit-based measures, such as morpheme error rate (MER), syllable error rate (SER) or character error rate (CER). For a fair comparison, different units should be transformed into a same unit set, word for example. CER is often used as word unit is not clear in some languages like Japanese and Chinese. WER is calculated by comparing the ASR results with a reference transcription.

$$WER = \frac{insertions + deletions + substitutions}{words\ in\ reference} * \ 100\% \qquad (2\text{-}8)$$

where the $insertions, deletions, substitutions$ are the numbers of incorrect hypotheses of the respective types, and their sum is divided by the total number of words in the reference.

As it is complicated to evaluate WER by performing ASR experiments, we need an intrinsic measure of the quality of an LM. The perplexity of an LM with respect to a test text S is the reciprocal of the geometric average of the probabilities of the prediction on S. When S contains N units, then the perplexity PP is calculated on basis of these units.

$$PP = 2^{H(S)} \qquad (2\text{-}9)$$

where the entropy is defined

$$H(S) = -P(S) \log_2 P(S) \qquad (2\text{-}10)$$

or more precisely, the cross entropy is computed

$$H(S) = -P(S) \log_2 P(S|model) \qquad (2\text{-}11)$$

As the $P(S)$ is unknown for the test data, a simple approximation is used

$$H(S) = -\frac{1}{N}\log_2 P(S|model) = -\frac{1}{N}\log_2 P(t_1 t_2 \dots t_N|model) \qquad \text{(2-12)}$$

As we can see the perplexity is different for the different basic units $\boldsymbol{t}$. To have a fare comparison between various units, the perplexity must be normalized in reference of a standard unit, word.

$$Normalized\ \ PP^{*} = PP^{\frac{N_t}{N_w}} \qquad \text{(2-13)}$$

where, N_t is number of units which is used for PP, and the N_w is the number of words (tokens) in the same data .

Both WER and perplexity are only comparable under the same lexicon set. For example, consider a random sequence of 100 units consisting of 4 evenly distributed unit set $\{A, B, C, D\}$. When double unit is selected as the basic unit set, there are 16 evenly distributed unit set $\{AA, AB, AC, AD, BA \dots\}$.

$A, B, C, D, B, C, A, A, B, D, C, D, B, A, C, B$ ... single unit-based sequence/
$AB, CD, BC, AA, BD, CD, BA, CB$ ... ... double unit-based sequence

When 10 among the 100 units are misrecognized, the unit error rate of single character is 10%. For the double unit-based sequence, the number of corresponding misrecognized units should be 5 to 10 among 50 units, so the error rate is between 10% to 20%.

Because they are evenly distributed, the single unit based entropy is 2, and the double unit based entropy is 4.

$$H(X) = -\sum_{i=1}^{4} P(x_i)\log_2 P(x_i) = -\sum_{i=1}^{4} \frac{1}{4}\log_2 2^{-2} = 2$$

For single unit set $X = \{A, B, C, D\}$ (2-14)

$$H(X) = -\sum_{i=1}^{16} P(x_i)\log_2 P(x_i) = -\sum_{i=1}^{16} \frac{1}{16}\log_2 2^{-4} = 4$$

For double unit set $X = \{AA, AB, AC, AD, BA \dots\}$ (2-15)

The corresponding perplexities are 4 and 16, but after normalization, by (2-13) they become same.

Theoretically, for an unlimited text corpus, perplexities based on various unit sets have similar results, as the probabilities of various units are equal as (2-16).

$$P(S) = P(t_1 t_2 \dots t_N) = P(r_1 r_2 \dots r_K) \quad (2\text{-}16)$$

In practical n-gram applications, in order to catch similar context, different units should use deferent dimension of n-grams. For example, if a word consists of 4 characters on average, word 3-gram has similar context to that of character 12-gram. However, the large n-gram dimension of short units is not efficient in storage space and decoding time compared to long units. On the contrary, the long units easily suffer from data sparseness which degrades coverage and perplexity for unseen data. So we need to search for the best lexicon set that is the longest units which maintain good coverage.

2.2 Review on concatenation approaches to lexicon optimization

A number of studies have been conducted to find an optimal lexicon. There are two main reasons. First, with a limited context of n-gram model, concatenating frequent sequences would improve the model. Second, long units may be better for matching than shorter units. Thus, length and frequency of the basic lexical units may have strong correlation with the performance of ASR systems [4][17]. Merging short and frequently co-occurred units are presumed to be effective for ASR in many languages like Japanese, Korean, Turkish, German, and so on.

2.2.1 Data-driven concatenation approaches

The simplest form of morpheme concatenation approach is based on merging the frequently co-occurred units. We call this as frequent morpheme sequence (FMS) concatenation method [19].

Some morphemes are very short and frequent, only consisting of one or two phonemes, and easily being confused in ASR systems. Merging short units would be an option. The *linguistic morpheme* structure of *agglutinative* languages is a stem followed by several suffixes. As the suffixes are very short compared to stems, merging all the singular suffixes followed a stem can make the stem and word-endings structure. This kind of sub-word structure is often adopted [14][89].

Mutual information (MI) and mutual bigram (MB) measures provide a criterion for merging candidates [6]. The mutual bigram is calculated with geometrical average of forward and reverse bigrams.

$$MI\big(t_i\ t_j\big) = \mathrm{P}(t_i\,, t_j) \log \frac{\mathrm{P}(t_i\,, t_j)}{\mathrm{P}(t_i)\mathrm{P}(t_j)} \qquad (2\text{-}17)$$

$$MB\big(t_i\ t_j\big) = \sqrt{\mathrm{P_f}(t_i|t_j)\mathrm{P_r}\big(t_j|t_i\big)} = \frac{\mathrm{P}(t_i\,, t_j)}{\sqrt{\mathrm{P}(t_i)\mathrm{P}(t_j)}} \qquad (2\text{-}18)$$

Note that $0 \leq$ MB ≤ 1 for every morpheme pair. These measures require the morpheme pairs to occur together, but also un-favor them if they occur in conjunction with other morphemes. Applications to several languages have reported successful reduction in WER [81][89].

2.2.2 Other statistical modeling for concatenation

A perplexity-based lexicon optimization method is proposed in [3]. Perplexity is calculated for basic phoneme units as a criterion for building new units. This approach reduced the vocabulary size by half with similar performance as the word based model.

A lexicon optimization method based on a formulation on the relationship between unit frequency, length, and recognition correctness is proposed in [4]. In this method the word correctness probability is modeled by the logit model as a function of word frequency and length. The formulation in this method is related with the WER results, but the relation is too general to capture specific morpheme pairs.

There are approaches which aimed to reduce OOV by transcribing the OOV words with sub-word lexical units [59][66]. And a sub-word unit based OOV detection method is proposed in [59]. In this method smaller sub-word units are applied to detect the OOV words and then other methods such as online learning of OOV words are used to improve ASR performance.

2.3 Unsupervised morpheme segmentation methods

Unsupervised lexical unit extraction methods have been successfully applied to many fields of natural language processing (NLP) applications [55][72]. Unsupervised probabilistic models are designed either to segment word units into sub-word (or *statistical morpheme*) units [73-77], or automatically selecting optimal units from a text.

2.3.1 Unsupervised lexicon extraction

The practical purpose of the segmentation is to provide a vocabulary of linguistic units that is smaller and generalizes better than a vocabulary consisting of words as they appear in text. Such a vocabulary could be utilized in statistical language modeling. Moreover, one could assume that such a discovered morpheme would correspond closely to *linguistic morpheme* of the language.

As the morphology discovery from untagged corpora is a computationally hard problem, in practice one must make some assumptions about the structure of words. The appropriate assumptions are somewhat language-dependent. For example, in English it may be useful to assume that words consist of a stem, often followed by a suffix and possibly preceded by a prefix. By contrast, in *agglutinative* languages, a word typically consists of a stem followed by multiple suffixes. And, only a few compound words and prefixes exist. Moreover, one may asks, whether a morphologically complex word exhibits some hierarchical structure, or whether it is merely a flat concatenation of stems and suffices.

Many existing morphology discovery algorithms focus on identifying prefixes, suffixes and stems, i.e., assume a rather simple inflectional morphology. D´ejean [44] addressed the problem of finding the list of frequent affixes for a language rather than attempting to produce a morphological analysis of each word. He identified possible morpheme boundaries by looking at the number of possible letters following a given sequence of letters, and then utilizes frequency limits for accepting morphemes. Goldsmith [46] focused on *stem + suffix* languages, in particular Indo-European languages, and tries to produce output that would match as closely as possible with the analysis given by a human morphologist. He further assumed that stems form groups that he calls *signatures*, and each signature shares a set of possible affixes. He applied a minimum description length (MDL) criterion for model optimization. These approaches consider only individual words without regard to their contexts or their semantic content. In a different approach, Schone and Jurafsky [48] utilized the context of each term to obtain a semantic representation for it using LSA. The division to morphemes is then accepted only when the stem and *stem + affix* are sufficiently similar semantically. Their method is shown to improve on the performance of Goldsmith's *Linguistica* on CELEX, a morphologically analyzed English corpus.

In the related field of text segmentation, one can often obtain morphemes. Some studies remove spaces from text and try to identify word boundaries utilizing e.g. entropy-based measures. Word induction from natural language text without word boundaries is also studied in [45], where MDL-based optimization is used. de Marcken [42] studied the problem of learning a lexicon, but instead of optimizing the cost of the whole corpus, he started with sentences with spaces kept. Utterances are also analyzed in [115] where optimal segmentation for an utterance is sought so that the compression effect over the segments is maximal. The compression effect is measured in what the authors call Description Length Gain, defined as the relative reduction in entropy. Brent presents a general, modular probabilistic model structure for word discovery [37]. He uses a minimum representation length criterion for model optimization and applies an incremental, greedy search algorithm which is suitable for on-line learning such that children might employ.

2.3.2 Segmentation based on MDL and MAP

Most of the previous works on statistical segmentation is based on the transitional probability of unit sequences. Sharon Goldwater et al [108] proposed a Bayesian framework for unit selection, this method is based on the probabilistic distribution of independent n-grams sequences [108]. A standard method using a probabilistic generative model is maximum likelihood (ML) estimation. But ML estimation easily leads to a problem of overfitting that it favors the hypothesis that fits the observed data very well, but generalizes very poorly to new data. In Bayesian learning, the effect of the likelihood can be counter-balanced with a prior distribution that favors simpler hypotheses. Simpler hypotheses do not fit the observed data so well, but will generalize more to new data. By considering both the likelihood and prior in determining the posterior probability of each hypothesis, Bayesian learning naturally avoid the problem of overfitting.

The trade-off between overfitting and generalization will depend on exactly how the prior is defined. Several methods have been used to define priors in previous Bayesian learning. The most well-known framework is known as minimum description length (MDL), and is exemplified by the work of de Marcken [42]. MDL has been used successfully in areas of language acquisition as well [46-59]. The basic idea behind MDL is to define some encoding scheme that can be used to encode the corpus into a more compact representation. In word segmentation, for example, description length is defined by the entropy, as well as the number of lexical items in a binary representation. With appropriate choices for the lexical items and binary representations (with shorter representations assigned to more frequent words), the length of the corpus could be reduced by replacing each word with its binary code. The minimum description length principle states that the optimal hypothesis is the one that minimizes the combined length, in bits, of the hypothesis itself (the codebook) and the encoded corpus. Although MDL can in principle produce good word segmentation results, there are no standard search algorithms for the optimal solution, and it is often difficult to design efficient model-specific algorithms. For example, Brent and Cartwright [116] were forced to limit their analysis to a very short corpus (about 170 utterances) due to efficiency concerns. In later research, Brent developed another Bayesian model for word segmentation with a more efficient search algorithm. He named this system Model-Based Dynamic Programming (MBDP-1) [37].

A cost function can be defined from MDL principle which simultaneously measures the goodness of the representation and the model complexity. Including a model complexity term generally improves generality by inhibiting overlearning, a problem especially severe for sparse data. Sequential splitting is applied and a batch learning algorithm is utilized.

In this work, it is assumed that words consist of lengthy sequences of *segments*. This model is suitable for languages with *agglutinative* morphological structure. The *segments* are called *statistical morpheme*s (or *morphs*) and at this point no distinction is made between stems and affixes.

A follow-up formulation by Creutz replaced the MDL formulation by a probabilistic maximum a posteriori (MAP) formulation, called the *Morfessor* system. It can be shown that, MDL and MAP are equivalent with some approximations and produce the same result [50].

The task is to induce a model of language from a raw text data in an unsupervised way. The model of language (*M*) consists of units and their various properties, and the goal is to find the optimal model which maximizes the following probability

$$argmax\ P(M|corpus) = argmax\ P(corpus|M)P(M) \qquad (2\text{-}19)$$

where

$$P(M) = P(properties) = M!\ P(\ properties(t_1) \ldots properties(t_N)) \qquad (2\text{-}20)$$

The properties can simply be frequency, length, or some linguistic and phonetic attributes [55].

$$P(M) = M!\ P(\ freq(t_1) \ldots freq(t_N)\ ) \cdot \prod_{i=1}^{N} [(1 - P(\#))^{length(t_i)} \cdot P(\#) \cdot \prod_{j=1}^{length(t_i)} P(c_j^{t_i})] \qquad (2\text{-}21)$$

2.4 Discriminative language model

Discriminative language modeling is also investigated recently; it utilizes alternative (negative) samples into account as well as the correct (positive) samples [30]. Thus, discriminative training estimates model parameters that discriminate well between different classes. In ASR framework, positive samples are the correct transcriptions and negative samples are the erroneous candidate transcriptions. Discriminative acoustic and language models utilize these samples to optimize an objective function that is directly related to the system performance. The discriminative LM approaches extracts relevant features, and train the weights on samples of ASR results [29-34]. The discriminative method proposed in this thesis adopts the similar idea of feature extraction and training, and apply it to the lexicon optimization.

Discriminative LM enhances the predictive ability of the probabilistic model itself rather than optimizing the basic units [29]. Discriminative LM can be complementary to the baseline generative LM [31]. A baseline language models is used in the first-pass recognition to generate the ASR transcriptions (lattices or the N-best lists), and discriminate these transcriptions into feature vectors and then re-rank the competent hypotheses.

Several other kinds of discriminative estimation of LMs has also been proposed in recent years. Jelinek [110] proposed an acoustic–sensitive language model whose parameters are estimated by minimizing $H(W|A)$, the expected uncertainty of the spoken text W given A, the acoustic sequence. Stolcke [58] experimented with various discriminative approaches, including MMIE, with mixed results. In the method of Stolcke et al. an “anti-LM”, estimated from weighted N-best hypotheses of a baseline ASR system, was used with a negative weight in combination with the baseline LM. Chen et al. [50] presented a method based on changing the trigram counts discriminatively, together with changing the lexicon to add new words. Kuo et al. [35] used the generalized probabilistic descent (GPD) algorithm to train a relatively small LM which minimizes string error rate on the DARPA Communicator task. This is an instance of the widely-known minimum classification error (MCE) training. Banerjee et al. [117] used a LM modification algorithm in the context of a reading tutor that listens. Their algorithm first uses a classifier to predict what effect each parameter has on the error rate, and then modifies the parameters to reduce the error rate based on this prediction.

2.5 Lexicon optimization in various languages

A number of studies have been conducted for many *agglutinative* languages such as

Japanese, Turkish, Korean, as well as other highly inflectional languages like Finnish, German, Estonian, Arabic, Hungarian, Thai, and so on. Sub-word unit based approaches and different optimization methods are reported for these languages. Among the highly inflectional languages, *agglutinative* languages preserve a relatively clear multi-layer morphological structure compared to other *fusional* languages such as Indo-European languages. Thus, the linguistic units in *agglutinative* languages can be extracted with a high accuracy, and their advantage can be compared with other units.

2.5.1 Lexicon optimization for Turkish

K. Hacioglu, et al [88] compared two major sub-word segmentation methods, morphology-based and data-driven methods. A morphological analyzer based on two-level-morphology is used for extraction of morphological (linguistic) morphemes, while the Morfessor program discussed in Sub-section 2.3.2 is used to extract *statistical morpheme*s. Some data-driven approaches such as mutual probability measure are used to avoid splitting highly frequent units, and improved ASR results are reported.

Various sub-word-unit based ASR systems are investigated for Turkish language by Murat Saraclar et al [89]. Word, *statistical morpheme*, *linguistic morpheme* units are compared in ASR applications. A stem-endings unit based approach is investigated by segmenting words into most frequent roots and endings, and this method outperformed other units. Furthermore, the sub-word units are also used for OOV detection by Maria Parada and Issam Bazzi [59][63].

Kemal Oflazer et all [104] investigated multi-layer sub-word segmentation by splitting words into stem and ending, if this is difficult, then split to syllables. This kind of segmentation incorporated with the phonological considerations has promising results.

Discriminative LM is applied for Turkish language by Ebru Arisoy, Brian Roark, and Murat Saraclar at al [31]. Morphological sub-word-unit based features like root n-gram and inflection group n-gram are incorporated, and the best result reduced WER by absolute 1% .

2.5.2 Lexicon optimization for Japanese and Korean

There is no word delimiter in Japanese language, and the supervised and unsupervised segmentation methods are applied for word extraction. Concatenative approaches are also applied based on frequency, likelihood, and correlation criteria. L. M. Tomokiyo and K. Ries [118] investigated a concatenation approach which merges the smallest syllable units based on the perplexity criterion.

R. K. Ando and Lillian Lee [119] investigated a mostly unsupervised lexicon segmentation based on the fact that there are three types of Japanese characters of kanji, hiragana, katakana, and changes in character type often indicate word boundaries.

Unsupervised lexicon extraction methods are applied to segmentation of Japanese by M. Nagata [52] who proposed a self-organized method which segment words from a small number of basic words and a large amount of raw text corpus. This method effectively augments the initial basic words with a re-estimation procedure. D. Mochihashi et al. [120] proposed a Bayesian model for fully unsupervised word segmentation, and outperformed other methods when applied to Japanese and Chinese languages.

Lexicon optimization approaches are also widely investigated Korean language. O.W. Kwon, K. Hwang et al [82] and Y. H. Park et al [84] investigated morpheme-based or pseudo-phonemes-based ASR approaches and several concatenation methods based on length and frequency attributes incorporated with some morphological and phonological properties. O.W. Kwon [83] investigated morpheme and syllable based ASR systems and a holistic application of several concatenation methods. By utilizing the length and frequency, morpheme units are merged by using rule-based and statistical concatenation methods.

2.5.3 Lexicon optimization for other inflectional languages

M. Creutz and K. Lagus [56] proposed an unsupervised morpheme segmentation approach for the inflectional language. They published the Morfessor program based on MAP framework as discussed in Sub-section 2.3.2, and applied it to many inflectional languages such as Turkish, Estonian, Arabic, and Hungarian, and improved ASR performances are reported [58].

There are many inflectional languages such as German [95], Arabic [96], Czech [102], and Thai [101] reported to have utilized morpheme or morpheme-like sub-word units to reduce OOV and improve ASR performance. Concatenation methods based on statistical measures and rule-based morphological and phonological criteria are applied to all these languages.

2.6 Conclusion

In this chapter, we reviewed conventional lexicon optimization approaches based on several different measures. Data-driven concatenation approaches are simple and shown to have effectively improved ASR performance. Unsupervised segmentation approaches are also used to extract basic units which are reported in some inflectional languages.

We have also reviewed discriminative LM which extracts relevant features, and train them on samples of ASR results. This method is improved and applied to the discriminative lexicon optimization addressed in this thesis.

Chapter 3

Morpheme segmentation and baseline ASR systems for Uyghur language

Uyghur language is an *agglutinative* language in which words are formed by suffixes attaching to a stem (or root). Because of the explosive nature in word vocabulary of the *agglutinative* languages, several sub-word linguistic unit sets are extracted.. As there are no delimiters for sub-word units, rule-based or statistical methods are used to segment words into smaller morphological units such as morpheme, syllable, phonemes and *statistical morpheme*s. Furthermore, on these various units, baseline ASR systems are built based on Julius platform [7][8]. WER results and lexicon sizes on various units based language models are compared. Here, morpheme is referred to any of prefix, stem, or suffix.

3.1 Uyghur language and morphological units

Uyghur belongs to the Turkish language family of the Altaic language system. At present, Uyghur is written in Arabic scripts with some modifications. There are 32 phonemes in Uyghur: 8 vowels and 24 consonants. One phoneme is recorded by one character. Sentences in Uyghur consist of words, which are separated by space or punctuation marks. Uyghur words consist of smaller morphological units without any splitter between them. Table 3.1 shows an example of various morphological unit sequences of word, morpheme and syllable and corresponding Japanese translation.

Table 3.1 Example of morphological units of Uyghur language.

Uyghur sentence	Müshükning	kəlginini	korgən	chashqan	hoduqup	qachti.
in Japanese	ねこが	きたのを	みた	ねずみ(が)	おどろいて	にげた
morpheme sequences	Müshük+ning	kəl+gən+i+ni	kor+gən	chashqan	hoduq+up	qach+ti
morphemes in Japanese	ねこ-が	き-た-の-を	み-た	ねずみ-(が)	おどろい-て	にげ-た
syllable sequences	Mü+shük+ning	kəl+gi+ni+ni	kor+gən	chash+qan	ho+du+qup	qach+ti
in English	The mouse who saw the cat coming was startled and escaped.					

The morpheme structure of Uyghur word is ***"prefix + stem + suffix1 + suffix2 + …"***. A stem (or root) is followed by zero to many (longest is about 10 suffixes or more) suffixes. A few words have a prefix (only one) in the head of a stem, and only 7 (difficult to find more) prefixes are used in this research. 108 suffix types are defined and collected, according to their semantic and syntactic functions, which can be extracted to 305 surface forms. The surface realizations of the morphological structure are constrained and modified by a number of language phenomenon such as insertion, deletion, phonetic harmony, and disharmony (vowel assimilation, vowel weakening [14][15]). Suffixes that make semantic changes to a root are derivational suffixes. Suffixes that make syntactic changes to a root are inflectional suffixes. A root linked with the derivational suffixes becomes a stem. So the root set is included in the stem set. Some examples are shown in Table 3.2. Sometimes the words "stem" and "root" are used without distinguishing. To keep the versatile nature of language, we keep different segmentation forms of a same word in our training corpus.

Table 3.2 Example of morpheme segmentation to stems or roots.

stems	root+ suffixes
oqutquchi (teacher)	oqut (teach) + quchi (er) {suffix}
yazghuchi (writer)	yaz (write)+ghuchi (er)
hesablinish (calculated, considered)	hesab (calculus)+la+n+idu, hesab+lan+idu

Syllable forms in Uyghur language have a relatively strict structure, the general format is "CV[CC]" (C stands for consonant, V stands for vowel)[15]. As a lot of foreign words are imported, new syllable formats are added such as "CCV[CC]" from some European languages, and "CVV[C]" from Chinese. Table 3.1 shows the syllable example.

3.2 Rule-based segmentation of morphological units

Uyghur morpheme segmentation is a basic part of the comprehensive effort of the Uyghur language corpus compilation. There are several ways of inducing morphemes. Some are rule based using some knowledge about the specific language such as morphological rules, stem list dictionary, and suffix list. Others are based on statistical models trained on a training corpus. The predefined morpheme units actually suffer from phonological and morphological changes. So the morpheme segmentation methods, either rule based or statistical model based, need to consider the phonological and morphological changes.

In addition to the supervised morpheme segmentation methods which need a manually prepared training corpus, there are unsupervised approaches which use only a raw text corpus to extract morphemes [40-57].

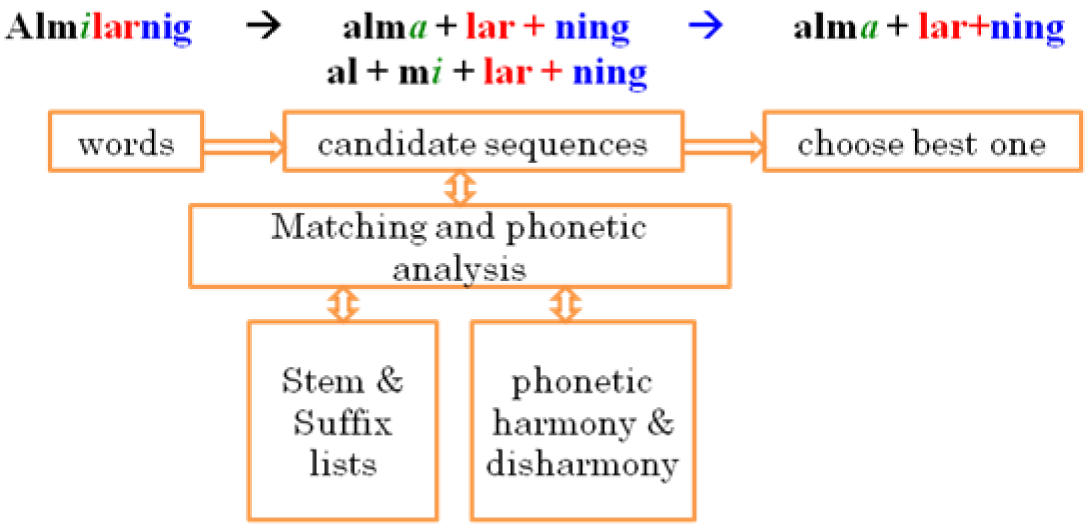


Figure 3.1 Flow chart of rule-based morpheme segmentation.

3.2.1 Inducing morphemes based on rules

Figure 3.1 shows the flow chart of the rule-based segmentation process. A stem-centered segmentation method is proposed in this research. The stem in Uyghur language remains fairly unchanged after suffixation compared to the suffixes which suffer from complicated changes of the surface forms. The surface realizations of morphological constructions are constrained and modified by a number of phonetic rules such as vowel and consonant harmony and vowel weakening [15]. We can manually summarize the phonetic rules, and implement them for the rule-based morpheme segmentation.

Forward and backward matching algorithm can be applied for the segmentation of words. At first, stems are collected manually from a dictionary [107]. To clarify, the root is the smallest independent meaningful unit; a stem is formed by attaching derivational suffixes to a root. Derivational suffixes change roots semantically while inflectional suffixes change syntactically. Therefore, stem list includes the roots as well. However, ambiguity exists for the changes, for nouns that become verbs or vice versa. Table 3.3 shows some examples of a root which is changed syntactically or semantically with different suffixes.

Table 3.3 Example of functional changes with suffixes.

morphemes	root+ suffixes
ish+ lesh	(work), become a verb
ish+ci	(worker), become a new stem
ish+ni	(the work), can only be an object

To prevent over-segmentation and secure the semantic identity of a word, stems are preferred than roots. So a complete stem list is important for the rule-based segmentation. The stem list provides the basis of segmentation, as the stem and suffix boundary is chosen as the primary target of segmentation. About 38,500 stems are collected which consists of almost all the common stems except for domain specific words and rarely used words.

After a word is separated into stem and word-endings (or combined suffixes), the segmented morphemes could further be segmented to singular suffixes by using singular suffix list, and by applying the phonetic rules which is necessary for recovering to standard surface forms. A relatively complete suffix list was obtained in an unsupervised way by applying these stems to a lexical corpus containing about 200,000 words. The training process was accomplished mostly by forward matching algorithm. A suffix list of compound and single suffixes are also extracted. From the extracted suffix, 325 singular suffixes with a standard form of about 108 types are verified by manual checking, and about 5880 compound suffixes are automatically selected by segmenting to their singular components. Furthermore, new compound suffixes are added automatically when the segmenter is trained on a new lexical corpus.

A matching algorithm is used to segment a candidate word using the stem list and the suffix list. When there are different segmentation results, the one with the longer stem is chosen to be the optimal, as the stem is preferable to root. For instance, for the word "atamning", segmentation results can be "at+am+ning", "atam+ning". Only semantic or context analysis could find out that the second one is correct. Choosing a longer stem decreases risk of incorrect segmentation, and a less number of morphemes may decrease the ambiguity and confusion.

3.2.2 Phonetic rules in Uyghur language

When the morphemes are combined each other to form a word, the surface forms of the morphemes often change around the boundary, according to some phonetic rules. The phonetic rules incorporated in the morpheme segmentation tasks discussed in this research is harmony and disharmony (final vowel change (weakening)) of phonemes.

1) Phonetic disharmony

When certain stems linked with some suffixes, the last vowel of the stem “a” or “ä” is replaced by two other vowels “i” or “e”; this phenomena is called final vowel change (or weakening, or disharmony) in Uyghur language. In the stem-suffix structure a word, when the last syllable of a stem is accentuated, two neighboring accents impact on each other and cause weakening on the former one. Phonetic disharmony is a complex phenomenon which is not fully predicted according to some rules [107].

From a text corpus collected from newspapers and text books, we extracted about 18,000 words, and vowel weakening is observed in about 12% words. As we do not know when the weakening happens, it should be checked for every candidate word. Below are examples of final vowel change.

“maktipi=maktap+i” ,	(somebody’s school.)
“adimi=adam+i”,	(person from somewhere.)

In this research, the method of solving the vowel weakening is to recover the weakened syllable. As we do not know which syllable is weakened, our method is to check one by one by recovering certain vowels. After a candidate word is segmented to syllables, find letters “i” and “e” which may have been weakened, replace them separately with “a” and “ä”. Then recovered words can be segmented by forward and backward matching algorithm. Several different segmentation results may be obtained. The stem can be over-segmented to a shorter stem and non-morphemes. For example, the word “almisi (apple’s)” can be segmented to three different results:

“almisi = alma + si,”

“almisi = al (take) + misi,”
“almisi = almas (diamond) + i.”

In these examples, first and third are correct segmentations with different meaning, and only by semantic or context analysis can determine the correct segmentation, we prefer a longer stem. Because of the recovery process while dealing with vowel weakening, different segmentations may happen. For example word “almilarning (apples’)” can be segmented to two segmentation results:

“almilarning = al (take)+milarning”   before recovery,
“almilarning = alma (apple)+larning”   after recovery.

In this situation, again we choose a longer stem as the preferred one. In the meantime, the suffixes analysis may also contribute to choose the correct one. For the new words, mostly imported from other languages, which are not in the stem list, segmentation is carried out according to suffixes only, and incorrect segmentation may be produced, especially when the vowel weakening is happened.

2) Phonetic harmony

Phonetic harmony is the harmony of vowels and consonants in the interface of the morpheme boundaries according to their pronunciations. There are two types of phonetic harmony in Uyghur language: consonant harmony and vowel harmony. Phonetic harmony is the basic controlling rule in the root-suffix linkage and syllable linkage. It happens at the interface of stem and suffix, and can be used to choose the correct surface form of a suffix. There are different forms of a same suffix in Uyghur language. There are four types of harmonization which caused different surface forms of morphemes as shown in Figure 3.2.

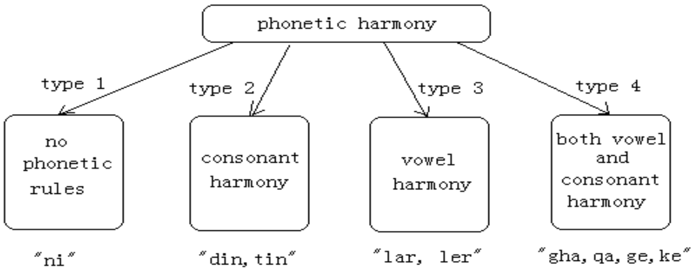


Figure 3.2 Phonetic harmonies in Uyghur language.

Type1: This kind of suffix has only one fixed form. For example, “ni, ning”, for the stem “adam” (person)

“adamning=adam+ning (correct)”
“adamni=adam+ni (correct)”

Type2: Consonants at the interface of morphemes must be harmonized according to the phoneme type of surd or sonant. For example, “din, tin”

“adamdin=adam+din (correct)”
“adamtin=adam+tin (wrong)”

Type3: Vowels at the interface of morphemes must keep harmonized according to the articulation point of vowels. For example, “lar, ler”

“adamlar=adam+lar (correct)”
“adamler=adam+ler (wrong)”

Type4: In this type, morphemes are harmonized according to both the type2 and type3, for example, “gha, qa, ge, ke”

“adamge=adam+ge (correct)”
“adamgha=adam+gha (wrong)”
“adamqa=adam+qa (wrong)”
“adamke=adam+ke (wrong)”

By analyzing harmonized surface forms, we can remove incorrect suffix segmentations.

3.2.3 Rule based segmentation results

We implemented a morpheme segmenter based on a list of lexical unit sets: word vocabulary, stem vocabulary, and suffixes vocabulary. Our corpora contain 38,500 stems, 305 singular suffixes, and about 5880 word-endings. We selected 18,400 words from the text corpus for the evaluation, and split them to morphemes. After manually checking the segmentation result, we estimate the accuracy of segmentation is 92%.

However, this rule-based method has only a limited applicability, for the training and test sets are based on only a word list, and the phonetic rules are only used to choose the correct segmentation result.

3.3 Morpheme segmentation based on statistical model

Another Uyghur morpheme segmenter has been developed based on statistical model. Our primary goal is to catch the different forms of stem, not root. This will expand the size of stem vocabulary, but is more convenient for analyzing semantic and syntactic context of words.

3.3.1 Corpus preparation and probabilistic modeling

A text corpus of 10025 sentences and their manual segmentations are prepared, details are in Table 3.4. These sentences are collected from general topics, unrelated. More than 30K stems are prepared independently and used for the segmentation task.

Table 3.4 Statistics of manually prepared morpheme text corpus.

	tokens	vocabulary
word	139.0k	35.4k
morpheme	261.7k	11.8k
character	936.8k	
sentence	10,025	

For a candidate word, all the possible segmentation results are extracted in reference for both stem and suffix, and their probabilities are computed to get the best result.

At first, a word is split into two parts, a stem and a combined suffix (word endings or stem endings), and several possible stem-suffix pairs are obtained. Then, the suffix is segmented into singular-suffixes, and each combined suffix may have several different singular-suffix segmentations.

There are several problems in the segmentation as in Table 3.5. First is the morphological change, which is deletion and insertion. Second is the phonetic harmony and disharmony (weakening or disharmony) [15] which cause different surface forms of a same morpheme. All morphemes should be recovered to standard surface forms in order to accurately train the probabilistic model. Third is the ambiguity.

Table 3.5 Examples of problems in morpheme segmentation.

segmentation example	problems
(1) almini = alma+ni, almiliring = alma+lar+ing	weakening
(2) oghli = oghul + i , kaspi = kasip + i	deletion
(3) qalmaytti = qal + may + [t] + ti, binaying=bina+[y]+ing	insertion
(4) yurttin = yurt + tin, watandin = watan + din	phonetic harmony
(5) hesablinidu = hesab+la+n+idu, hesab+lan+idu, berish = bar(go/have)+ish, bər(give)+ish	ambiguity

Generally, an intra-word bigram method based on the following probabilities is used, and the identification of stem-suffix boundary is the most important part in this segmentation

$$P(stem-suffix\ boundary)=\begin{cases}P(stem,firstSuffix)=\frac{C(stem,firstSuffix)}{C(stem)}\\P^{'}(stem)P(anySuffix|stem)\quad for\ smoothing\end{cases}\tag{3-1}$$

in which

$$P^{'}(stem)=\frac{C(stem)}{stemToken+stemVocabulary}\tag{3-2}$$

$$P(anySuffix|stem)=\frac{C(stem,anySuffix)}{C(stem)}\tag{3-3}$$

where $C(t_i)$ is the frequency of t_i

We used different solutions for the morphological and phonetic problems. For insertion, we added the inserted phoneme to the subsequent suffix, and formed a new surface form of the same suffix type. For deletion, because it happens in the stem only, a list of deleted stems are learned from the training corpus. For the phonetic rules, we have to recover every of the 305 suffix forms into one of their original standard forms of 108 types, so that the suffixes are accurately learned with their standard forms.

3.3.2 Segmentation results

We split the corpus to the training corpus of 9025 sentences, and the test corpus of 1000 sentences. The results of coverage and segmentation accuracy are shown in Figure 3.2. Word coverage is 86.85%. Morpheme coverage is 98.44%. The morpheme segmentation accuracy is **97.66%** which is the percentage of the exact match of all morphemes in automatic segmentation compared with manual segmentation.

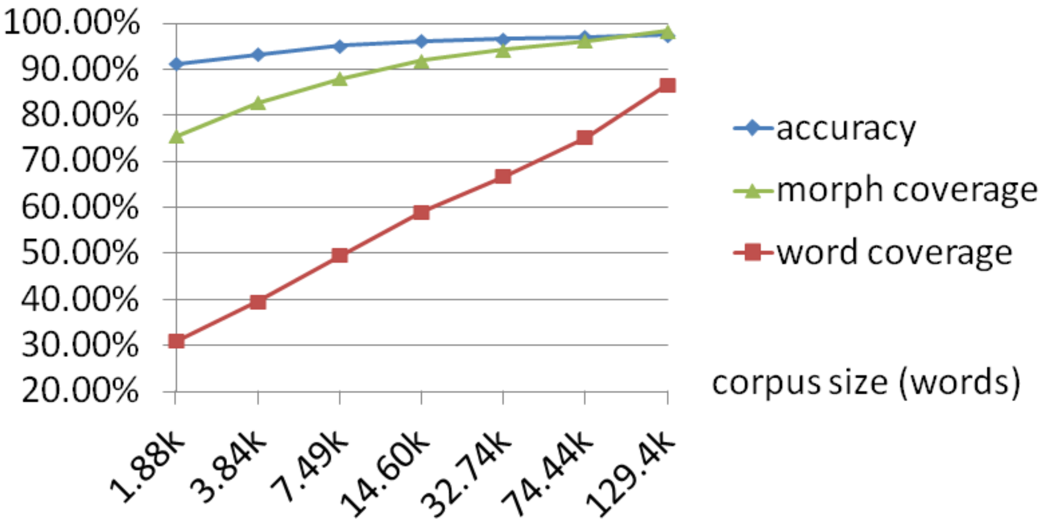


Figure 3.2 Morpheme segmentation accuracy and coverage based on morpheme and word units.

Generally two kinds of ambiguity exist in our segmentation. One is because of the definition of the stem set; the other is because of the sound harmony.

In the last example in Table 3.5, the weakened stem (“bar” or “bər”) has a same surface form when attached by some suffixes. Both words are frequent words, and both results have high probabilities, but only the most probable one is produced from this tool.

3.3.3 Syllable segmentation

Syllable is another clear unit in Uyghur language. The Uyghur words in the general syllable format CV[CC] consist of about 99.1% of all words in our corpus. The words in the format of foreign syllables are about 0.6%. Except the misspelled words (around 0.3% by estimation), all words can be correctly segmented with our rule-based syllable segmenter. There may be ambiguities for a few words which are in the foreign syllable format. There are no changes in the surface forms after syllable segmentation.

3.4 N-gram language models on various units

Lack of resource is one of the biggest problems for Uyghur language processing. From various publications, we prepared a raw corpus of about 630k sentences. They are from general topics such as novels, newspapers, and books (history, science...). This corpus is cleaned by removing all duplicated sentences, as it was a collection of different sources and may have many copies of same content. We segmented the texts in this corpus separately to morphemes and syllables, and built trigram language models based on three different units: word, morpheme, and syllable. Among the various units, only morpheme is the meaning bearing unit. *Statistical morpheme*s, syllables, and characters are relatively random sequences. All punctuation marks are removed in the following experiments to make the coverage and perplexity consistent in the LM evaluation and ASR evaluation.

Changes in the surface forms especially by the phonetic rules cause problems for practical applications of morpheme based LMs. In Uyghur language, speech is recorded as pronounced. When a word is segmented, if there is phonetic disharmony, usually it is recovered to the standard surface format. We keep the surface forms of morphemes same as in the words, thus the words can be recovered simply by connecting morpheme sequences without any changes, as shown in Table 3.6.

Table 3.6 Examples of surface form changes.

words	morphemes (co-articulated)	morphemes with standard surface forms
teghi	tegh+i	tagh+i
almiliringiz	almi+lir+i+ngiz	alma+lar+i+ngiz

These may cause some ambiguity in morphemes, but do not degrade segmentation accuracy. In this way, the statistical properties of coverage and perplexity are compared with n-gram language models.

In order to preserve the word boundary information, we add a symbol of word boundary between syllables and characters, or label the morpheme as prefix, stem, or suffix, as shown in Table 3.7. For syllable and character units, a word boundary symbol is added between syllables or characters in the place of word boundary. For morphemes and *statistical morpheme*s, the prefix and suffix are labeled, nothing added to stem. In this way, the word sequences can easily be recovered morpheme sequences by simply reconnecting them together.

Table 3.7 Example of inserting word boundary information for various units.

Unit forms	People are unaware of the event.
Word sequence	Kishilər wəqədin bihəwər qaldi.
Morpheme sequence	Kishi _lər wəqə _din bi_ həwər _qaldi.
Syllable sequence	Ki+shi+lər _ wə+qə+din _ bi+hə+wər _ qal+di.

Trigram models are built on various units, respectively; Kneser-Ney smoothing is adopted. Unknown word model is <UNK> used, and words appeared only once are considered as unknown.

As a test corpus, 11888 sentences are held out with the character size of 1460.8k, Table 3.8 shows statistics of the test corpus. From the statistics, we can see a word unit is segmented into about 1.88 morphemes and 2.73 syllables, and 1.03 *statistical morpheme*s (Section 2.3.2) on average. The remaining 620K sentences are used as a training set. Figure 3.3-3.6 and Table 3.9 show the results of vocabulary size and coverage. Table 3.10 shows trigram perplexity comparison. The result shows that the morpheme-based language model performs comparably to the word-based language model with a much smaller size and better coverage.

Table 3.8 Statistics of test corpus for LM evaluation.

units	word	morph	syllable	statistical morpheme	character
tokens	217k	409k	593k	189k	1.4M
vocabulary	47k	15.3k	3.6k	43.4k	33

Table 3.9 Comparison of trigram LMs based on various linguistic units.

training corpus (sentences)			77.5K	155K	310K	620K
subset of learning corpus L			1/8	1/4	1/2	1/1
Vocabulary size	word	Freq>0	149,347	222,729	329,370	480,067
		Freq>1	63,501	97,461	149,054	227,101
	morph.	Freq>0	40,403	61,146	93,627	144,765
		Freq>1	17,823	25,145	37202	57768
	syllable	Freq>0	7049	9813	13,948	20,088
		Freq>1	4084	5180	6976	9846
#tokens	word		1,453,870	2,904,037	5,806,217	11,587,471
	morpheme		2,748,350	5,487,041	10,965,894	21,869,762
	syllable		3,987,209	7,959,562	15,906,966	31,744,522
character token			9,818,903	19,601,528	39,178,682	78,187,496
unigram coverage (%)	word		91.47	93.71	95.47	96.71
	morpheme		98.76	99.02	99.25	99.40
	syllable		99.87	99.90	99.93	99.95
bigram coverage (%)	word		53.07	58.64	64.56	71.10
	morpheme		89.60	92.07	94.08	95.77
	syllable		97.18	98.15	98.80	99.26
trigram coverage (%)	word		16.93	21.07	26.79	34.88
	morpheme		64.48	70.32	75.82	81.14
	syllable		82.84	87.24	90.80	93.69
perplexity	word		12856	6857	3689	1929
	morpheme		91.27	77.43	66.20	56.99
	syllable		26.13	24.12	22.52	21.228
normalized perplexity by word	word		12856	6857	3689	1929
	morpheme		5078.4	3706.8	2747.6	2059.8
	syllable		7699.7	6150.7	5076.3	4316.5

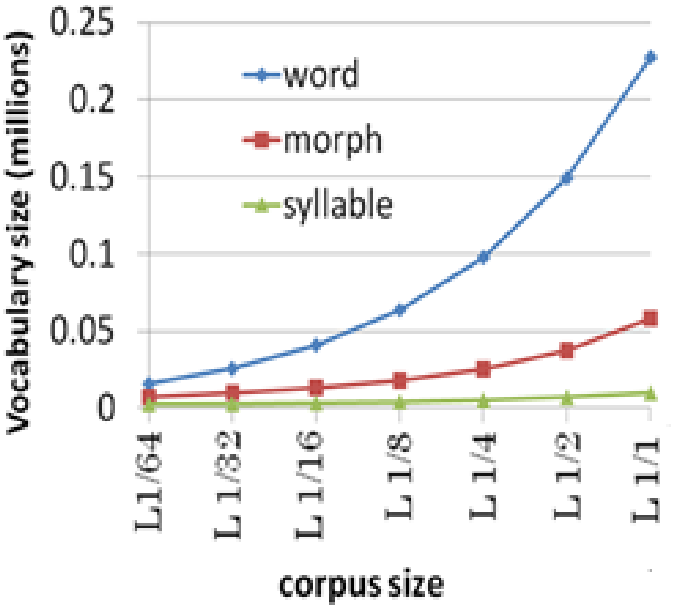


Figure 3.3 Vocabulary size of various units.

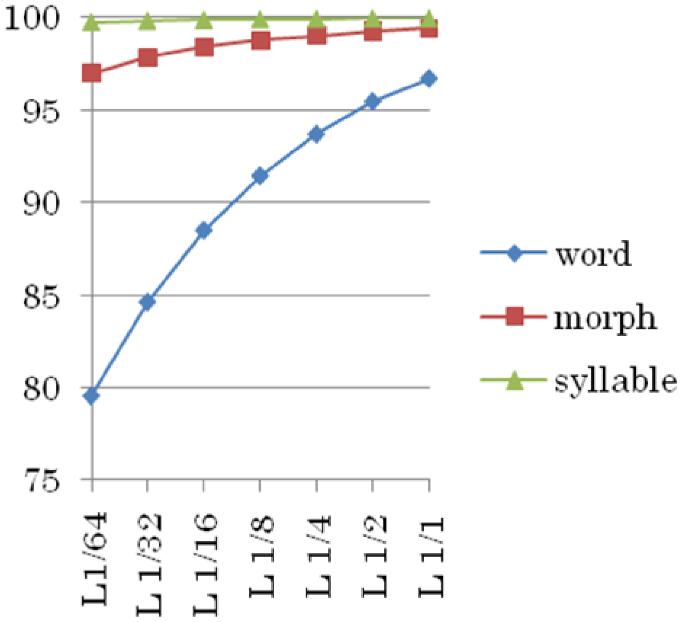


Figure 3.4 Unigram coverage of various units (%).

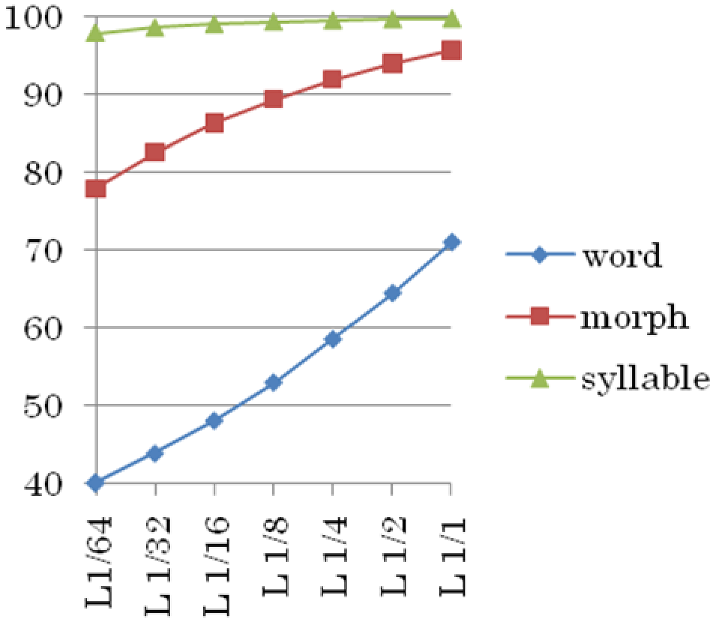


Figure 3.5 Bigram coverage of various units (%).

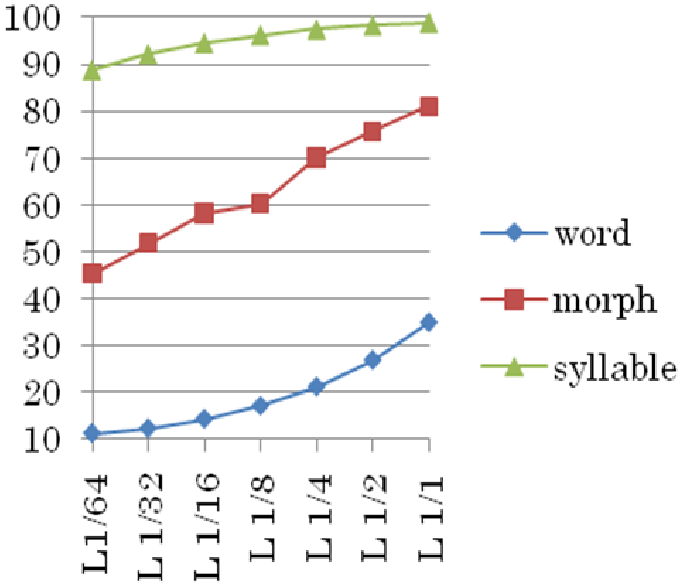


Figure 3.6 Trigram coverage of various units (%).

Table 3.10 Perplexity by trigram models of various units.

training corpus	perplexity			perplexity normalized by words		
	word	morpheme	syllable	word	morpheme	syllable
L1/64	125554	178	37.3	125554	18137	20684
L 1/32	53295	136	32.3	53295	10813	13865
L 1/16	25196	110	28.7	25196	7263	10067
L 1/8	12856	91	26.1	12856	5078	7680
L 1/4	6858	77	24.1	6858	3707	6150
L 1/2	3689	66	22.5	3689	2447	5076
L 1/1	1929	57	21.2	1929	2060	4316

Then, we compare n-gram models with different n sizes. Because of the memory limitation, we can only calculate until 5-gram for word and morpheme units, 6-gram for syllable unit, and 10-gram for character unit. To compare the results, the perplexity is normalized in reference to the word unit. Table 3.11 and Figure 3.7 show the result.

Table 3.11 Normalized perplexity of n-gram models of various units.

units	word	morpheme	syllable	character
vocabulary size	227.1k	57.7k	9.8k	33
1-gram	29,302	482,973	110,014,618	30,014,487,856
2-gram	3039	6294	168,482	140,025,078
3-gram	1929	2060	4316	4,498,647
4-gram	1754	1318	3349	2,17,874
5-gram	1700	1091	1901	29,051
6-gram			1425	9186
7-gram				4743
8-gram				3113
9-gram				2397
10-gram				2032

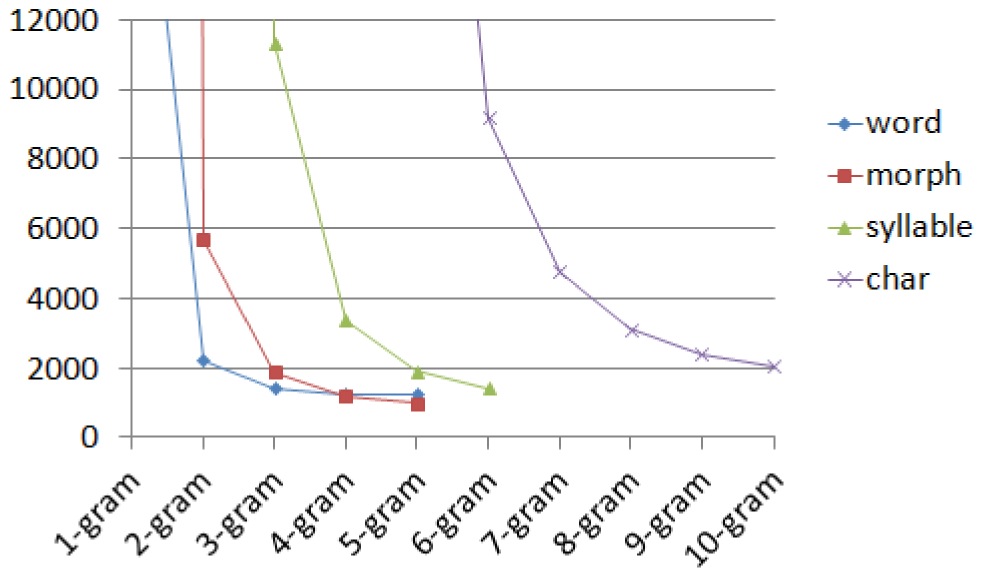


Figure 3.7 Perplexity comparisons on various n-grams.

The results show that similar context modeling converges on similar perplexity results. For example word bigram has similar perplexity with character 10-gram, like we have discussed in Sub-section 2.1.4. Longer units easily have smaller perplexity with smaller n size which may both reduce time and space complexity compared to smaller units. Among them the *linguistic morpheme*-based model performs better in coverage and perplexity compared to other units.

.

3.5 Baseline ASR systems based on various units

We then built an ASR system using the language models, on the basis of Julius system. Julius is open-source large-vocabulary continuous speech recognition (LVCSR) software for researchers and developers. Various acoustic models and language models are easily pluggable, and you can build various kinds of speech recognition systems by preparing your own models suitable for the task. It also adopts standard formats to handle other toolkits such as HTK, CMU-Cam SLM toolkit, and so on.

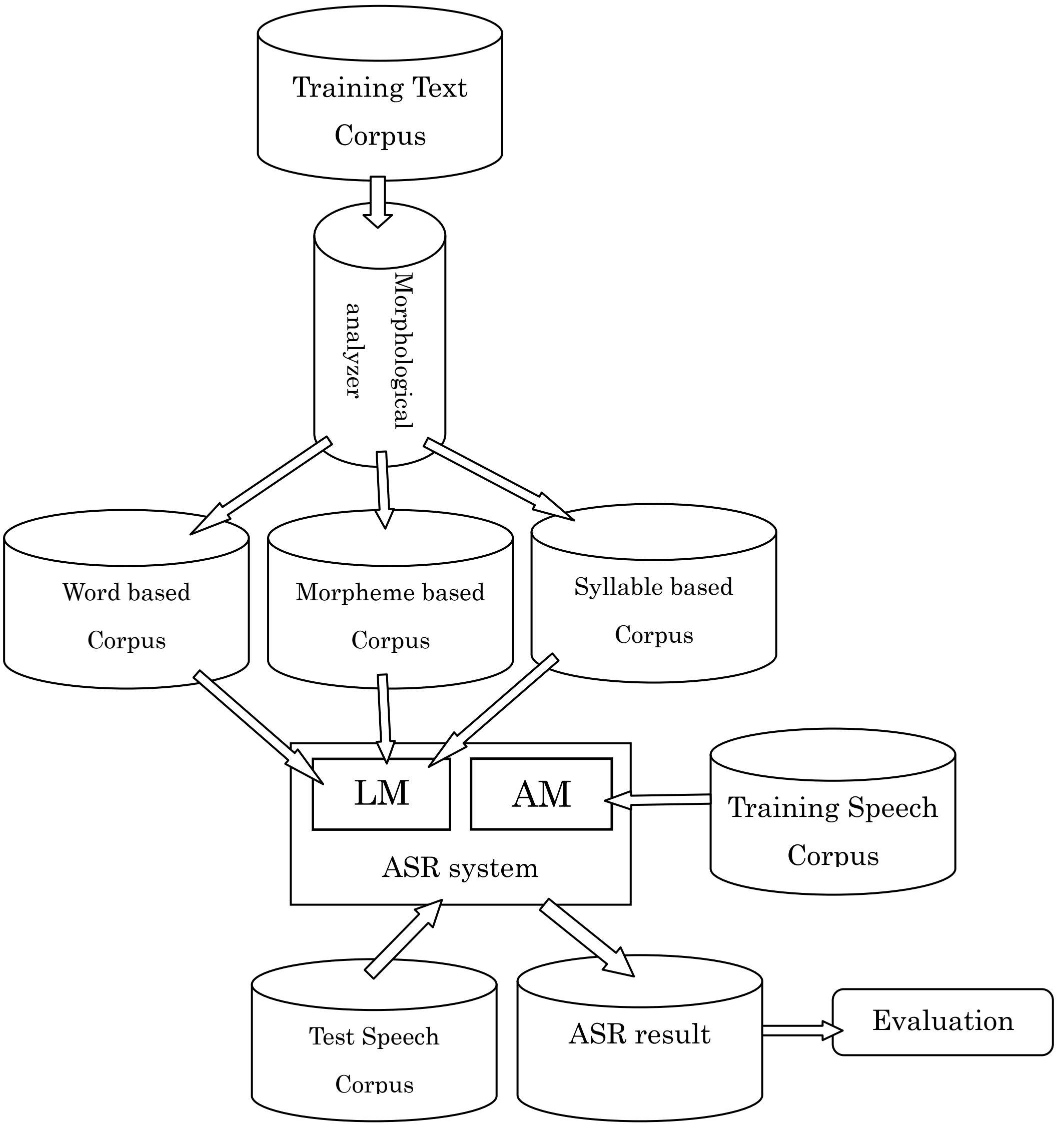


Figure 3.8 Process flow of construction of ASR system.

Figure 3.8 shows the ASR system construction process for various units. The training speech corpus is used to build an acoustic model which is used for all the experiments in this thesis. Language models are separately built on different units described in the previous Sub-section. ASR results based on various units are compared fairly with WER of the same word unit.

3.5.1 Uyghur acoustic model

A speech corpus of general topics is prepared to build an acoustic model for Uyghur language. This corpus is also used as the training data for lexical optimization addressed in this work. A held-out test data set is prepared from readings of newspaper articles. Specifications of the data sets are summarized in Table 3.12. Training speech corpus is recorded by 187 female and 166 male speakers, aged from 19 to 28. Test speech corpus is recorded independently by 13 female and 10 male speakers, aged from 22 to 28.

Table 3.12 Statistics of training and test speech corpora.

corpus	#unique sentences	#speakers	#utterances	word tokens	morpheme tokens	time (hours)
training	13,415	353	61,784	895,075	1,676,985	158.6
test	550	23	1250	14,660	27,609	2.4

There are 32 phonemes in Uyghur, 8 vowels and 24 consonants. One character corresponds to one phoneme, so there are 32 different characters, with one additional character which is actually a syllable segmentation mark. We used 34 basic phonemes including silence. HTK is used to build three-state HMMs. They are tri-phone HMMs with 3000 shared states and 16 Gaussian mixtures prepared for 34 Uyghur phones (8 vowels, 24 consonants, and 2 silence models). Acoustic features consist of 12 MFCCs, ΔMFCCs and $\Delta\Delta$MFCCs together with Δpower and $\Delta\Delta$power.

3.5.2 Comparison of various units with WER

For language modeling, the text corpus of 630K sentences described in section 3.4 is used. Several different LMs are built using the training corpus on various morphological units, and ASR results are compared. The word boundary symbol is added to all sub-word units. For n-gram modeling, the Kneser-Ney smoothing method is adopted. The cutoff threshold for n-gram ($n \geq 2$) is set to 1. Following models are prepared and compared.

①Word-based LM: 3-gram, 4-gram.

②Morpheme-based LM: 3-gram, 4-gram, 5-gram.

③Stem & word-endings (stem-suffix) LM; words are segmented into two parts: stem and combined suffix (word-ending, or stem-ending). In other words, all the singular suffixes are combined. The 3-gram model is trained.

④Syllable based LM: 3-gram.

The morphemes can be connected to words, so instead of using morpheme error rate (MER) we can conveniently use word error rate (WER) for fair comparison of all different units. The results are shown in Table 3.13.

The vocabulary of syllable-based LM is 6.58k and the syllable error rate (SER) is 28.73%. It is very difficult to recover words from syllable units, for it causes too many confusion. So we used an opposite way. We automatically segmented the word-based ASR result to syllables with our syllable segmenter, and calculated the SER, which is 15.42%, much better than the syllable based ASR result.

The stem word-endings based model is actually an optimization based on linguistic units. The singular suffixes are short and often occur together, so merging them will not increase the vocabulary size very much as shown in Table 3.13. We can reduce confusion, but this simple optimization method does not bring a large improvement.

Table 3.13 ASR error rates for different LMs.

LM names	WER (%)	MER (%)	vocabulary size	word perplexity	OOV rate
Word 3-gram	25.72	18.88	227.9k	2356	2.8%
Word 4-gram	25.93	19.02	227.9k	1734	2.8%
Morph 3-gram	28.96	22.73	55.2k	1733	0.3%
Morph 4-gram	27.92	21.64	55.2k	1244	0.3%
Morph 5-gram	29.31	22.98	55.2k	1144	0.3%
Stem & word endings 3-gram	28.13	21.69	82.6k	1633	0.5%

The results show that the word-based language model performs better than the morpheme based model. The similar tendency is observed in other inflectional languages such as Japanese, Korean, Turkish, German [81-103]. To have a low OOV rate and a reliable language model with the word unit, however, a very large training data set is needed. Even in the current task with a large training corpus, the OOV rate is 2.8%.

Moreover, the morpheme-based model can be expanded to a huge vocabulary while the vocabulary of the word-based model is limited to words observed in the training corpus.

The cutoff threshold also controls the lexicon size and ASR performance. Cutoff-F means that units with frequency less than F times are disregarded and treated as unknown. The cutoff threshold for n-gram (n ≥ 2) is set to 1. The ASR performance is compared in Table 3.14.

In the following chapters, we use the morpheme 4-gram model with cutoff-5 as the baseline because the difference from the cutoff-2 case is not statistically significant. The model with Cutoff-5 can reduce the lexicon size and decoding time of the ASR system, thus save space and time, without degrading the ASR performance very much.

Table 3.14 ASR results for different baseline units.

Baseline models		WER (%)	lexicon size	word perplexity	OOV
morph. 4-gram	cutoff-2	27.92	55.2k	1244	0.3%
	cutoff-5	28.11	27.4k	1303	0.7%
word 3-gram	cutoff-2	25.77	227.9k	2356	2.8%
	cutoff-5	26.64	108.1k	3162	4.4%

3.6 Conclusion

We propose a morphological unit segmentation method for the Uyghur language. During the design and implementation of the supervised morpheme segmenter, we manually segmented and standardized the Uyghur morphemes, especially for the suffixes. By collecting large text and speech corpora, we have obtained reliable statistics for Uyghur language on various units. We also built ASR systems based on different units. In the baseline ASR evaluations, the word-based model outperforms other linguistic sub-word unit based systems.. However, note that to have low OOV and a reliable language model with the word unit, a very large training data set is needed. Otherwise, the ASR performance would be degraded very much. This property is not good for applying ASR to various domains. We investigate a hybrid of word-based model and morpheme-based model to realize both a high coverage and good ASR performance. The morpheme-based LM provides a perspective of practical Uyghur language processing.

Chapter 4

Morpheme concatenation approach based on feature extraction from two layers of ASR results

4.1 Introduction

The goal of this study is to make compatible the vocabulary size of the morpheme unit and the accuracy of the word unit. The objective is to find word entries which reduce the WER with a minimum increase of the vocabulary size. In this study, we investigate approaches of selecting word entries by concatenating morpheme sequences, which would reduce WER. Specifically, we compare the ASR results of the word-based model and those of the morpheme-based model. From the aligned ASR results of the morpheme and word based systems, we extract critical samples in which the word unit is correctly recognized when the corresponding morpheme units are misrecognized, as in Table 4.1.

Table 4.1 Aligned ASR results based on morpheme and word units.

Uyghur sentence	Müshükning	kəlginini	korgən	chashqan	hoduqup	qachti.
morpheme sequences	Müshük+ning	kəl+gən+i+ni	kor+gən	chashqan	hoduq+up	qach+ti
ASR result (word)	Müshükning O	kəlginini O	korgən O	chashqan O	hoduqup O	qachti O
ASR result (morph.)	Müshük+ning O	kəl+gin+i+ni X	kor+gən O	chashqan O	hoduq+ip X	qach+ti O

Some data-driven approaches have been investigated with the word frequency basis or likelihood criterion [81-94]. However, these criteria are not directly related to WER. In this work, we extract useful patterns by aligning and comparing the ASR results by the morpheme-based model with those by the word-based model. These patterns are classified according to length, error frequency, and attribute of the units, and individually assessed in terms of their contribution for reducing WER. Specifically, we extract frequently misrecognized morpheme sequences which are correctly recognized by merging them to words; we then identify short and frequently misrecognized morphemes by separating them to stem and word-ending, and recombine them in all possible ways. Each individual feature and their combinations are investigated by generalizing them on the morpheme-based corpus. Figure 4.1 shows the optimization process.

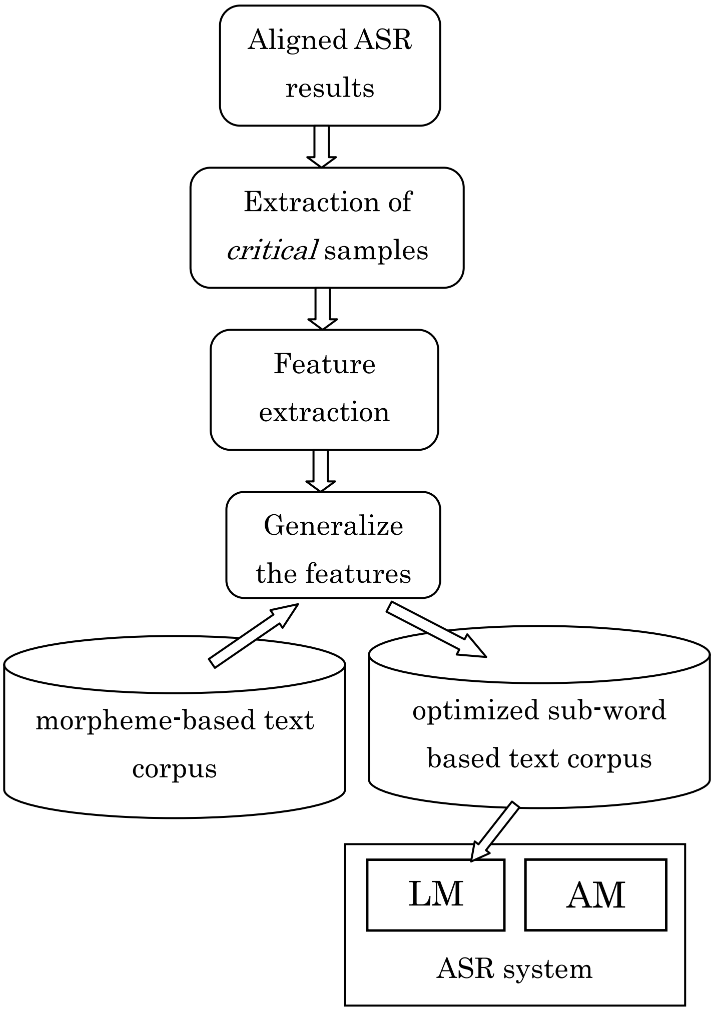

Aligned ASR results
Extraction of *critical* samples
Feature extraction
Generalize the features
morpheme-based text corpus
optimized sub-word based text corpus
LM
AM
ASR system

Figure 4.1 Flow chart of manual feature extraction and optimization approach.

4.2 Analyzing aligned ASR results

The speech corpus which was used for acoustic modeling is used for the analysis. The details of held-out and test sets are shown in Table 3.12 and Table 4.2. With this held out corpus, an acoustically closed ASR result is obtained.

Table 4.2 Comparison of held out and test speech corpora.

Unit forms	Morpheme 4-gram		Word 3-gram	
Speech corpus	Held out corpus	Test corpus	Held out corpus	Test corpus
1-gram coverage (%)	98.4	99.7	94.5	97.2
2-gram coverage (%)	94.5	96.4	75.5	71
3-gram coverage (%)	82	81.6	47.4	31.3
perplexity	82.8	52.2	2436	2497
WER (%)	29.61	28.11	26.80	25.77

The held out corpus covers a variety of phoneme sequences, so has more stem entries compared to the test set.

We compare the ASR results by the word-based model and those by the morpheme based model. We can classify the patterns related with the confusion into two categories: lexical properties and acoustic properties. The lexical properties related with the lexical unit selection include length (number of syllables) and attribute (stem or word-ending). The acoustic properties can be attributed to co-articulation effects [6]. These properties can be systematically analyzed with linguistic and phonological knowledge. However, instead of speculating the patterns of unknown results, it is convenient to directly observe the ASR results, and enumerate the problematic patterns. Thus, we identify major reasons in confusion for morpheme sequences in comparison with word sequences, as in Table 4.3.

Table 4.3 Main reasons of misrecognized morpheme sequences.

Main reasons for misrecognition	Examples (English translation)
Phonetic harmony or co-articulation	yigirmə-yigirmi (twenty), vottura-votturi (middle)
Confusion in frequent short stems with many derivatives	biz, vu, bash, yər (we, he/her, head, land)
Phonetic similarity	həmmə-əmma (all, but)
Ambiguity	uni-u+ni (he, him)
Too many suffix insertions	ish+lap+p+i+ish+ni

Among these patterns, the co-articulation problem caused by phonetic changes can be solved by recovering the morpheme sequences into word units. Similarly, we can extract other problematic morpheme sequences. A simple solution would be to extract all the problematic morpheme sequences and merge them into words. Our preliminary study showed that this approach works well, but it is difficult to cover all the erroneous words in the open test data. Therefore, we also explore a more generalized model.

In this work, the cause of the confusion by the morpheme-based model is attributed to three types of features: error frequency, length, and attribute (stem or word-endings). And these features can be manually defined and extracted.

4.2.1 Error frequency feature

First, as a simple method, we investigate the frequency of misrecognized morpheme sequences. We collect all the words which are misrecognized by the morpheme-based model, but correctly recognized by the word-based model. They are added to the lexicon with a threshold of the frequency of recognition errors. From the ASR results of training data, we collect the candidates of word entries whose error frequency is higher than twice.

$$\Phi_{freq}(w) = \begin{cases} \text{true} & if \;\; w \;\; misrecognized\; more\; than\; twice \\ \text{false} & otherwise \end{cases}$$

This method can be iterated to select more candidates. On each iteration, new candidates are extracted and added to the vocabulary, until few new candidates are found.

4.2.2 Length feature

Short units are easily confused in ASR and usually they are very frequent. Confusion in short morphemes can be reduced by merging and making them longer. There are many single-phoneme suffixes produced from our morpheme segmenter. To make them longer, all the short morphemes are merged to each other when they are neighbors or to the neighboring morphemes. Below is an example of the length feature.

$$\Phi_{length}(m_i) = \begin{cases} \text{true} & if\ \ length(m_i)\ \ is\ less\ than\ 2 \\ \text{false} & otherwise \end{cases}$$

While stem & word-ending models described in Section 2 are based on statistical co-occurrence of morphemes, the proposed method directly considers their effect on the ASR performance.

4.2.3 Attribute (stem or word-ending)

We also conduct a simple morphological analysis to find generalized features. From the aligned ASR results, we separate the morpheme sequence into two parts within the word unit boundary: stem and word-ending. Then we separately collect all misrecognized stems and word-endings based on their error frequency. As a result, short and frequent stems are typically extracted. These short stems have many derivatives, and are easily confused. The word-endings are also collected according to their error frequency so that they are connected with these short stems. A brief feature description is as follows.

$$\Phi_{stem}(st_i) = \begin{cases} \text{true}\ \ if\ stem\ \ st_i\ \ is\ \ misrecognized\ more\ than\ 10\ times, \\ \qquad\qquad and\ length(st_i)\ is\ less\ than\ 4\ syllables \\ \quad \text{false} \quad otherwise \end{cases}$$

$$\Phi_{word_{ending}}(we_j) = \begin{cases} \quad \text{true}\ \ if\ word\ ending\ \ we_j\ \ misrecognized \\ and\ in\ conjuncture\ with\ any\ short\ stem\ \ st_i\ , \\ \qquad\qquad \text{and}\ \Phi_{stem}(st_i) = \text{ true} \\ \quad \text{false}\ \ otherwise \end{cases}$$

These features are generalized by merging all possible combinations of stems and word-endings into words when both features are observed in the training corpus.

4.2.4 Combination of features and language modeling

In the above-mentioned methods, effective features are identified separately. Each feature is used to concatenate morphemes within the word boundary. A new designed unit set is used to build a new language model. However, when we apply all the above mentioned features together to build a new unit set, the new ASR system based on this optimized unit set is not accumulatively improving the ASR performance because of the feature redundancy. Instead of applying three features in one iteration, we separately apply the features step by step in several iterations. In this way, we can iteratively apply all above mentioned three features.

4.3 Experimental evaluation

The proposed methods are evaluated by applying to the Uyghur LVCSR task. The morpheme-based 4-gram language model is used and compared with the 3-gram word-based model.

The first method is based on the error frequency. From the training data, the words misrecognized more than twice are extracted, and added to the vocabulary. In Table 4.4, WER for the training and test data after two iterations are listed. When we extract misrecognized words from the test set, we found that only 50% of them are covered by the training data set. This simple method may not have generality, for it cannot include entries that are not in the training data. However, the method is very effective for reducing WER.

Table 4.4 Results of word selection based on error frequency feature.

Iterations	Baseline	First round	Second round
WER(%) on training data	31.95	28.62	27.01
WER(%) on test data	28.11	26.11	25.82
Vocabulary size	27.0k	40.3k	46.0k

In the second method, the morphemes consisting of single phoneme are merged to each other or to the previous morpheme. This simple method made 0.92% reduction of WER, as shown in Table 4.5.

In the third method, we separate the morpheme sequences into stem and word-ending, and merge them in all possible ways. This method made 1.36% decrease in WER from the baseline model.

Finally, we combine the above proposed method as shown in Table 4.5. The error frequency feature is taken after the second round. We confirm an accumulative effect. The final result here outperforms the word-based model result in Table 4.2, with a much smaller vocabulary size.

Table 4.5 Optimized ASR results using various features and their combined action.

Models	WER (%)	∆WER (%)	Vocabulary size
Morpheme-based baseline	28.11	-	27357
Error frequency feature	26.11	2.00	40376
Length feature	27.19	0.92	32881
Attribute feature	26.74	1.36	36333
Attribute feature+ Length feature	25.80	2.31	41257
Attribute feature+ Length feature+ Error frequency feature	24.89	3.22	56718

4.4 Conclusion

We have proposed a method for morpheme concatenation for effective LVCSR. Instead of analyzing linguistic or statistical property from the linguistic rules or text data, we analyze the ASR results and identify useful patterns by comparing the ASR results of word and morpheme units. We extracted problematic morpheme sequences based on several features: error frequency, length, and attributes. The concatenation method based on these features significantly reduced WER from the morpheme-based baseline model without a drastic increase of the lexicon size compared with the word-based model. The iterative optimization method improved the ASR performance accumulatively.

Chapter 5

Discriminative lexicon optimization approaches

5.1 Introduction

In the previous chapter, effective features are manually extracted to define a lexicon. In this chapter, we propose a machine learning approach to select word (or sub-word) entries which are likely to reduce the WER [16]. It is also realized by aligning and comparing the ASR results by the morpheme-based model with those by the word-based model. We describe each word by a set of features, and define an evaluation function with their weights. Then, the weights are learned to select "critical" word entries which generate different (probably correct) hypotheses from the morpheme-based units. This learning mechanism is applicable to any unseen words, or even sub-words. We investigate several discriminative models including Support Vector Machines (SVM) and Logistic Regression (LR) model.

The proposed scheme is applied to and evaluated in the large-vocabulary Uyghur ASR system. Several features are investigated and compared in terms of WER and the lexicon size.

5.2 Discriminative learning for lexicon optimization

The proposed discriminative approach to lexicon optimization is realized by comparing the ASR results by the morpheme-based model and those by the word-based model. The results are aligned by word with corresponding morpheme sequences. We assume each word is composed of one or more morphemes, and morpheme units do not cross word boundaries. An example is given in Table 5.1.

When these two ASR results are different, neither of them is correct in most cases (approximately 68%). In the remaining cases, in which either of the two hypothesis are correct, the word-based model gives much more correct hypotheses than the morpheme-based model (28.5% vs. 3.5%), as suggested by the results of Table 3.14 and Table 5.2. Therefore, a naïve method would be to pick up these "critical" word entries (e.g. “cheghinglarda” in the example of Table 5.1) to be added to the lexicon, and shown to be very effective in improving ASR performance. However, this method heavily depends on the training data set since it can select only entries observed there, and thus may not have generality.

Table 5.1 Example of ASR results of aligned morpheme and word units.

reference (word)	Yash	cheghinglarda	bilim	elishinglar ək	ker
reference (morph.)	Yash	chegh_ing_lar_da	bilim	el_ish_ing_lar ək	ker
ASR result (word)	Yash O	cheghinglarda O	bilim O	berishinglar ək X	ker O
ASR result (morph.)	Yash O	chegh_ing_da X	bilim O	el_ish_ing_laək O	ker O
in English	Study when you are young.				

Table 5.2 Percentage of critical samples.

Word ≠	Morph.	percentage
X	X	68.0%
O	X	28.5%
X	O	3.5%

Naïve method = error frequency sampling

Table 5.3 Example of feature extraction from samples.

w	$m_i\, m_j$…	word ≠ morph.
cheghinglarda	chegh_ing_lar_da	O X
$\Phi_{\mathrm{bigram}\_m_i\, m_j}(w) = \begin{cases} 1 & \text{if } morph.bigram\ (m_i\ m_j)\ exists\ in\ w \\ 0 & \text{otherwise} \end{cases}$		$y^i = 1$

5.2.1 Evaluation function of words with lexical features

In the proposed scheme, each word w is described by a set of features $\Phi(w) = (\Phi_s(w);\ s = 1, \dots k;\ \Phi_s(w) \epsilon \{0, 1\})$ of the constitute morphemes $(w = m_1 m_2 \dots m_f)$, and its desired value y $(y \epsilon \{0, +1\})$ is defined by the differences of ASR results of the two units as shown in Table 5.3. We assume that they are binary (+1 for true, 0 for otherwise).

Training sample pairs $\left(\Phi\left(w^i\right), y^i\right)$ are extracted independently for every word observation and its corresponding morpheme sequence. Given all training sample pairs $(\Phi\left(w^i\right), y^i\ ;\ i = 1, \ldots l\ )$, we feed them to the training scheme. In this work, we adopt and compare three different machine learning algorithms: perceptron, SVM, and LR.

For the perceptron algorithm, we define an evaluation function as a linear weighted sum of the features [29].

$$f(w^i) = \sum_s \Phi_s\left(w^i\right)\mathrm{\alpha_s} = \Phi(w^i)\ \alpha \qquad (5\text{-}1)$$

Here, α_s is a weight for the feature $\Phi_s\left(w^i\right)$. The above function indicates the potential importance of the word to be included in the lexicon, or how likely WER will be reduced by adding this word entry. Note that this function can be computed for any words or even sub-words consisting of morpheme sequences, so that we can select effective entries which would not be correctly recognized by the morpheme-based model.

5.2.2 Discriminative models for weight training

In the perceptron algorithm, the standard sigmoid function is introduced to map the above evaluation score to the 0-1 range.

$$g(w) = \frac{1}{1+e^{-f(w)}} \qquad (5\text{-}2)$$

$$g^{'}(w)|_{f(w)} = g(w)(1 - g(w)) \qquad (5\text{-}3)$$

Then, the weight vector is updated as:

$$\alpha = \alpha + \eta\ g^{'}(w^i)(y^i - g\left(w^i\right))\Phi(w^i) \qquad (5\text{-}4)$$

The learning rate parameter η is adjusted at every iteration to prevent excessive fluctuation. Here we simply reduce it by a factor of 10. The difference $(y^i - g(w^i))$ controls the direction of adjustment. This learning converges in several iterations, and we terminate at the third iteration in the experiments. This simple method is convenient to implement. However, this kind of adjustment (4) easily falls into a local optimum when incorporating a large number of features. On the other hand, the following SVM and LR methods are more sophisticated in dealing with a large dimension of features.

For the SVM and LR, we adopt a linear binary classifier [24-26]. We modify the formulation so that $y^i \epsilon \{-1, +1\}$ (+1 for true, -1 for false). Given the same set of training sample pairs $(\mathbf{\Phi}(w^i), y^i)$, both methods solve the following unconstrained optimization problem with different loss functions $\xi(\alpha;\ \boldsymbol{\Phi}(w^i), y^i)$:

$$min_{\alpha} \frac{1}{2}\ \alpha^T \alpha + C \sum_{i=1}^{l} \xi(\alpha;\ \mathbf{\Phi}(w^i), y^i) \qquad (5\text{-}5)$$

where C>0 is a penalty parameter. For SVM, the two popular loss functions are:

$$\max(1 - \ y^i \alpha^T \Phi(w^i), 0) \qquad (5\text{-}6)$$

and

$$|\max(1 - \ y^i \alpha^T \Phi(w^i), 0)|^2 \qquad (5\text{-}7)$$

The former is referred to as L1-SVM, and the latter is L2-SVM. In our experiments, we use L2-SVM.

The loss function for LR is:

$$\log(1 + e^{-y^i \alpha^T \Phi(w^i)}) \qquad (5\text{-}8)$$

which is derived from a probabilistic model. The SVM optimization is stopped at the tolerance of 0.1, and the LR training stopped at the tolerance of 0.001.

5.2.3 Unsupervised training scheme

The values of the weights $\alpha = \{\alpha_s\}$ are estimated based on the above described models using the training data set. The desired output y^i is defined as binary, corresponding to the CRITICAL_CASE in which the word-based model outputs a different hypothesis from the sequence generated by the morpheme-based model.

$$y^i = \begin{cases} +1 \ \text{ if } \ \text{CRITICAL\_CASE is true} \\ 0 \ \text{ or } \ -1 \quad \text{ otherwise} \end{cases} \tag{5-9}$$

Note that the above judgment does not refer the correct hypotheses. There are some cases in which the word-based model makes an error while the morpheme-based model generates a correct hypothesis as shown in the right-hand example of Table 5.2. However, the ratio of such cases among all differences is only 3.5% as shown in the previous Sub-section. We also introduce sample filtering as described in the next sub-section. The property of not using the reference transcripts makes the proposed training in an unsupervised fashion, so that we can make use of enormous un-transcribed speech data.

5.2.4 Filtering training samples

The unsupervised training essentially involves erroneous samples. The discriminative models especially the simple perceptron algorithm is not robust against noisy or outlier samples. Thus, we introduce filtering so that only reliable samples are fed to the training. Specifically, we selectively use the samples whose frequency of the CRITICAL_CASE is more than N times over the entire training data set. This is expected to be effective for discarding erroneous samples made by the word-based model because these errors are not frequent and consistent.

5.3 Lexical features

In this section, we list the lexical features considered in this work. We represent the candidate word as w_i, the corresponding morphemes as m_i, the stem as st_i, and the word-ending as we_i. A specific weight is estimated for each feature.

5.3.1 Word ID

This corresponds to a naïve method which matches only word entries. It also makes a constant feature for all word entries, i.e. always becomes 1.

$$\Phi_{\text{word\_}w_i}(w)=\begin{cases} 1 & \text{if } w = w_i \\ 0 & \text{otherwise} \end{cases}$$

Below is the example of the second word in Table 5.2.

$$\Phi_{\text{word\_cheghinglarda}}(cheghinglarda)=1$$

5.3.2 Morpheme length

Short units are easily confused in ASR and they are very frequent. Actually, there are many suffixes consisting of only one or two phonemes. Confusion in short morphemes can be reduced by merging and making them longer. The feature counts the length of the morphemes by the number of constituent phonemes.

$$\Phi_{\text{length\_L}}(w) = \begin{cases} 1 & \text{if } \exists\ m_i \quad length(m_i) \leq L \\ 0 & \text{otherwise} \end{cases}$$

Below are examples of this feature.

$$\Phi_{\text{length\_1}}(cheghinglarda) = 0$$
$$\Phi_{\text{length\_2}}(cheghinglarda) = 1$$

5.3.3 Morpheme N-gram

Here, we focus on typical morpheme entries and their bigram patterns. A specific weight α_s is estimated for each unigram or bigram entry.

$$\Phi_{\text{unigram\_}m_i}(w)= \begin{cases} 1 & \text{if } morph.m_i\ exists\ in\ \ w \\ 0 & \text{otherwise} \end{cases}$$

$$\Phi_{\text{bigram\_}m_i\,m_j}(w) = \begin{cases} 1 & \text{if } morph.bigram\ (m_i\ m_j)\ exists\ in\ w \\ 0 & \text{otherwise} \end{cases}$$

Below are examples of the morpheme sequence appeared in Table 5.2.

$$\Phi_{\text{unigram }\_lar}(cheghinglarda) = 1$$

$$\Phi_{\text{bigram}\_ing\ \_lar}(cheghinglarda) = 1$$

5.3.4 Morpheme Attributes

We also categorize morphemes into stems and word-endings which are a sequence of suffixes.

$$\Phi_{\text{stem\_st}_\text{i}}(w) = \begin{cases} 1 & \text{if } st_i\ exists\ in\ w \\ 0 & \text{otherwise} \end{cases}$$

$$\Phi_{\text{word}-\text{ending\_we}_\text{i}}(w) = \begin{cases} 1 & \text{if } we_i\ exists\ in\ w \\ 0 & \text{otherwise} \end{cases}$$

Below are examples appeared in Table 5.2.

$$\Phi_{stem\_chegh}(cheghinglarda) = 1$$

$$\Phi_{word-ending\_ing\_lar\_da}(cheghinglarda) = 1$$

5.4 Lexicon design and extension to sub-word units

These features are then generalized to all words in the text corpus for LM training. If the candidate word is judged as CRITICAL_CASE or the evaluation function $g(w)$ is larger than a threshold (=0.5), we select it to be included in the lexicon, with cufoff-5 threshold applied. Otherwise the word is left as morpheme units. The threshold of 0.5 is a natural choice since $g(w)$ is a sigmoid function.

Furthermore, the method can be applied to sub-words, which are composed of morpheme sequences within a word. Specifically, we try to search for sub-word entries $(m_1\ m_2 \ldots)$ that satisfy the condition $g(m_1\ m_2 \ldots) > 0.5$. The search is exhaustively done from the beginning of all words by concatenating the succeeding morphemes while the above-mentioned condition is satisfied. If the condition is not met, the search is re-started there.

An overview of the proposed scheme is depicted in Figure 5.1. Baseline ASR systems are prepared with both morpheme-based units and word-based units, and they are applied to a large-scale speech database that was used for acoustic model training. We can use even un-transcribed speech data, as the proposed learning scheme is realized in an unsupervised manner. Using the data set, the proposed discriminative function is trained. Then, it is applied to the LM training database, which is much larger in text size. Once the lexicon is prepared by incorporating the word or sub-word entries, 4-gram LM is trained again and the entire test data are decoded again using the new model.

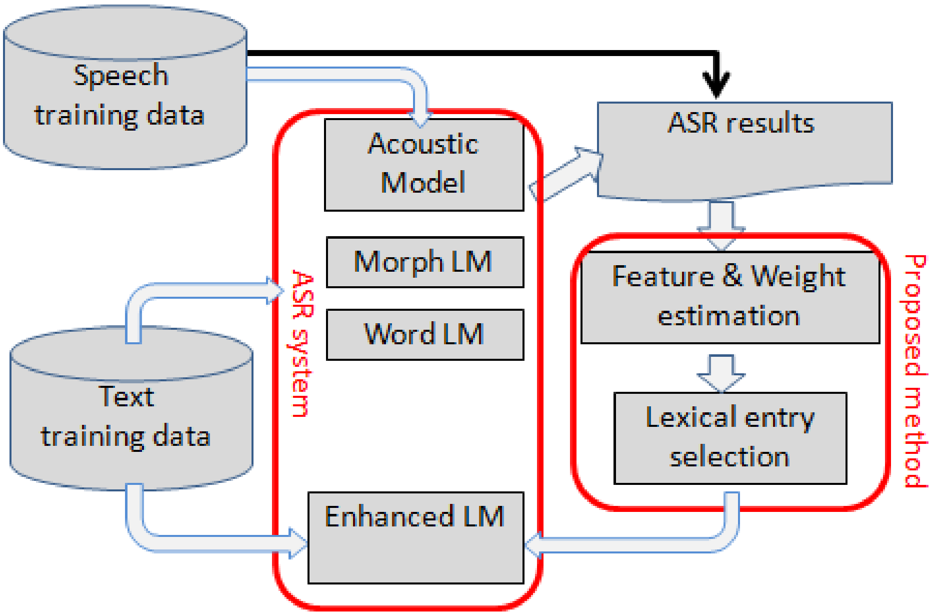


Figure 5.1 Overall flow of the proposed discriminative training scheme.

5.5 Experimental evaluations

The method has been implemented and applied to our Uyghur LVCSR system described in Chapter 3.

5.5.1 Effect of sample filtering

First, we investigate the effect of sample filtering described in Section 5.2.4. In this experiment, we use the morpheme unigram features applied to the word-level selection. The WERs obtained by changing the threshold (N) values are listed in Table 5.4. We can see that removing outlier (possibly erroneous) samples of only one occurrence is effective for the perceptron algorithm, but not so much for the SVM and LR. This results show that SVM and LR are trained more robustly and reliably against outlier samples. Based on the results, we set N=0 for SVM and LR, and N=2 for the perceptron in the following experiments.

Table 5.4 Effect of sample filtering threshold (unigram feature).

threshold		N=0	N=1	N=2	N=3	N=4
perceptron	WER (%)	26.69	25.93	25.87	26.18	26.28
	lexicon size	104.5K	90.2K	74.8K	63.6K	55.3K
LR	WER (%)	25.99	25.57	25.91	25.93	26.01
	lexicon size	102.4K	91.2K	79.9K	70.1K	62.4K
SVM	WER (%)	26.05	26.03	25.93	25.93	26.00
	lexicon size	103.4K	94.6K	83.7K	73.5K	65.4K

5.5.2 Comparison of features

The effects of individual features listed in Section 5.3 are compared in Table 5.5. In this experiment, we use the perceptron algorithm applied to the word-level selection. Although the length feature alone is not so effective because of its simplicity, all other features lead to significant improvement from the baseline morpheme model (WER=28.11%), and the accuracy is comparable to the best word-based model with Cutoff-2 (WER=25.77%; the WER difference among these methods are not statistically significant). Note that the lexicon size of the enhanced morpheme-based model is much smaller than the word-based model (230K with Cutoff-2), and still expected to give a broad coverage. Combinations of these features are also explored, but little additional gain is obtained due to the redundancy of these features.

Table 5.5 Comparison of features in word selection (perceptron).

Feature	WER (%)	lexicon size
(cf.) morpheme 4-gram (cufoff-5)	28.11	27.4K
word	26.18	35.8K
length =1	27.07	32.4K
length ≤ 2	27.08	35.1K
unigram	25.87	74.8K
bigram	25.99	67.3K
stem	26.10	92.7K
word-ending	26.20	92.1K
stem & word-ending	25.96	82.3K
(cf.) word 3-gram (cutoff-2)	25.77	227.9K

5.5.3 Result of sub-word selection

Then we generate sub-word lexical entries by using the morpheme N-gram features. Here we compare the morpheme unigram and bigram features. The dimension of the unigram features is 17K and that of the bigram is 53K. The result in Table 5.6 shows that this method reduces both WER and the lexicon size significantly. The proposed optimization is more effective when conducted thoroughly in the sub-word level than the word level. The sub-word-based model trained with the bigram feature outperforms the best word-based model in accuracy with the lexicon size of one fourth. From the results we can see that the SVM and LR methods are more effective especially with a large dimension of bigram features.

Table 5.6 Results of sub-word selection.

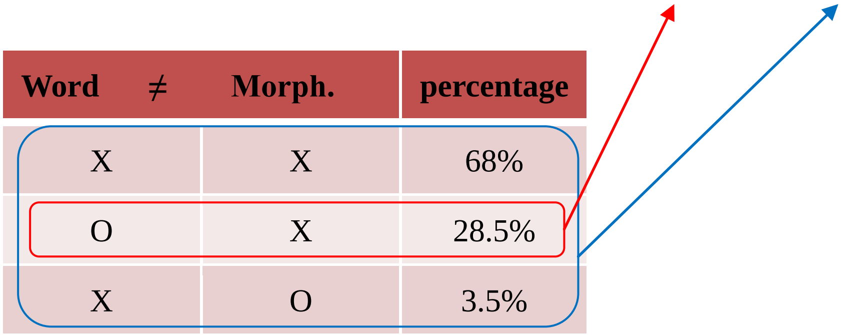
Units		Word		Sub-word	
Features		unigram	bigram	unigram	bigram
perceptron	WER (%)	25.87	25.99	25.96	25.27
	lexicon size	74.8K	67.3K	40.7K	49.9K
LR	WER (%)	25.99	25.75	25.77	24.87
	lexicon size	102.4K	85.4K	44.0K	65.8K
SVM	WER (%)	26.05	25.86	27.05	24.61
	lexicon size	103.4K	80.1K	34.7K	55.1K

5.5.4 Reference experiment with supervised learning

We also conducted a supervised training experiment for reference. We compare the best results by using the bigram feature applied to sub-word level selection. In this case, we collect positive samples $(y^i = +1)$ from those in which the word-based model gives a correct hypothesis and the morpheme model made an error. The all remaining samples are regarded as negative samples $(y^i = 0\ or - 1)$. This means that the number of positive samples is decreased to 28.5%, compared with the unsupervised training. The results listed in Table 5.7 show that the supervised training does not perform better than the unsupervised training. Instead, the unsupervised training outperforms for the SVM and LR. As previously mentioned, in the majority cases (68%) of the CRITICAL_CASE in which the two models give different results, both results are incorrect. We reason, however, that there are many informative samples among them in which the word-based model gives "better" hypotheses than the morpheme-based model.

Table 5.7 Supervised experiment results of sub-word selection with bigram feature.

training schemes		supervised	unsupervised
perceptron	WER (%)	25.55	25.27
	lexicon size	49.7K	49.9K
LR	WER (%)	25.34	24.87
	lexicon size	46.3K	65.8K
SVM	WER (%)	25.42	24.61
	lexicon size	45.1K	55.1K



Word ≠	Morph.	percentage
X	X	68%
O	X	28.5%
X	O	3.5%

5.5.5 Comparison with statistical methods

Finally, the proposed model is compared with the following two conventional statistical models.

Co-occurrence frequency

A simple model based on statistical co-occurrence is built by merging the frequent morpheme sequences (FMS). Specifically, we count the morpheme bigram co-occurrence frequency $C\left(m_i\, m_j\right)$, and concatenate them if the frequency is higher than a threshold. The concatenation process is repeated to a sequence of morphemes, just like the proposed sub-word model, except that the concatenation can be made even across the word boundaries.

Mutual bigram

Another statistical measure is mutual bigram (MB) [6]. It is calculated as a geometrical mean of forward and reverse bigram probabilities as below. Here the $\mathrm{C}(-)$ represents the occurrence counts of unigram and bigram patterns.

$$MB\left(m_i\, m_j\right) = \sqrt{\mathrm{P_f}(m_i|m_j)\mathrm{P_r}\left(m_j|m_i\right)} = \frac{\mathrm{C}(m_i\;\, m_j)}{\sqrt{\mathrm{C}(m_i)\mathrm{C}(m_j)}} \qquad \text{(5-10)}$$

Note that these methods including the proposed model concatenate a sequence of morphemes, but the criterion of the concatenation is different. The results by varying respective threshold values are listed in Tables 5.8 and 5.9. It is observed that the proposed method is significantly better than the best results by these methods. Moreover, the tuning of the threshold values for these methods are not so straight-forward, depending on the task and database, while our proposed method does not have any sensitive parameters.

We also investigate the combination of the proposed method with the statistical method. Here, we adopt a tandem approach; first apply the proposed perceptron based discriminative method, and then apply the best MB-based method. Lexicon entries are added by each step. The results are summarized in Table 5.10. The simple combination results in drastic improvement in accuracy, 1% absolute compared with the best word-based model. The result shows the discriminative model has a synergetic effect with the statistical model.

Table 5.8 Result of frequent morpheme sequence (FMS) method.

Threshold	2000	2500	3000	3300
lexicon size	57.1K	50.7K	44.8K	42.3K
WER (%)	27.02	26.63	26.68	26.76

Table 5.9 Results of mutual bigram (MB) method.

Threshold	0.030	0.035	0.040	0.045	0.050	0.06
lexicon size	69.1K	60.0K	53.3K	47.0K	41.9K	36.1K
WER (%)	25.83	25.61	25.60	25.79	25.80	26.07

Table 5.10 Combined result of perceptron and mutual bigram methods.

Methods	WER (%)	lexicon size
mutual bigram	25.60	53.3K
bigram sub-word (perceptron) + mutual bigram(threshold=0.04)	24.75	67.8K
bigram sub-word (SVM) + mutual bigram(threshold=0.06)	24.21	63.2K

The major results are summarized in Table 5.11, with different cutoff threshold values for the proposed model.

Table 5.11 Summary of discriminative optimization results.

Models		WER (%)	lexicon size	word perplexity	OOV rate
baseline morpheme (cutoff-5)		28.11	27.4k	1303	0.7%
baseline word (cutoff-2)		25.77	227.9k	2356	2.8%
stem & word endings (cutoff-2)		28.13	74.5k	1633	0.5%
mutual bigram (best case)		25.60	53.3k	1886	0.7%
sub-word bigram feature (SVM)	cutoff-2	24.64	101.2k	1258	0.7%
	cutoff-5	24.61	55.1k	1394	0.9%

5.7 Conclusion

We have proposed a novel discriminative approach to lexicon optimization for *agglutinative* languages. It adopts the same scheme as the conventional statistical approach which starts with the morpheme-based model and search for effective word or sub-word entries to be added. However, the proposed discriminative learning is directly linked to the improvement of ASR accuracy. It can take into account not only linguistic constraint but also acoustic-phonetic confusability in ASR. In the experimental evaluations, the proposed method achieves the best accuracy in comparison with other statistical methods, resulting in a significant gain from the baseline morpheme-based model and the word-based model without a large increase in the lexicon size. We also made comparison of discriminative models of SVM, LR, and perceptron, and found that SVM and LR are more effective than the perceptron algorithm. The proposed learning scheme is realized in an unsupervised manner, so it can be applied to a large amount of un-transcribed data.

Chapter 6

Comparison of lexicon optimization methods for Uyghur LVCSR

6.1 Introduction

We have described several lexicon optimization methods based on the concatenation of morpheme units. We have proposed *automatic* and *manual* methods which can extract problematic morpheme sequences from the ASR results. In this chapter, we compare them with segmentation methods and in both manual and automatic manners.

A manually prepared training corpus of the morpheme sequence is necessary for the supervised segmentation of *linguistic morphemes*. The *linguistic morphemes*, which are defined as the smallest meaning bearing units, are not optimal for ASR. In order to have better ASR performance, sub-word units are selected with various optimization methods as we have discussed in previous chapters. These optimized sub-word units are derived from concatenation of *linguistic morphemes*. In Chapter 4 and Chapter 5, we presented both manual and automatic lexicon optimization methods by comparing ASR results of word and morpheme based LMs. And these methods greatly reduced the WER and the lexicon size compared to the word based ASR system.

In unsupervised segmentation methods, words are split into morpheme-like units from a raw text corpus [46-58]. One of the most well-known methods is Morfessor. As they are not always strictly meaning-bearing units, sub-word linguistic information may be lost. Many works show that the *statistical morpheme*s improve the ASR performance [86-90].

We also investigate a morpheme concatenation method based on the statistical framework which can accomplish concatenation directly from the morpheme-based text corpus. We adopt the maximum a posteriori (MAP) model used for the Morfessor tool [57], This optimization method based on the predefined *linguistic morphemes* preserves word and morpheme boundaries. Table 6.1 is the brief description of the approaches compared here.

Table 6.1 Lexicon optimization approaches.

	manual learning	automatic learning
Segmentation	*linguistic morpheme* (baseline; Chapter 3)	statistical morpheme (Morfessor)
Concatenation	manual feature extraction from word and morpheme based ASR results (Chapter 4)	discriminative learning approach (Chapter 5)
		Statistical concatenation approach (this Chapter)

Figure 6.1 shows the pipeline of the lexicon optimization process. First, we investigate the sub-word segmentation approaches for linguistic and *statistical morpheme*s. Second, concatenation approaches are investigated based on *linguistic morpheme*s.

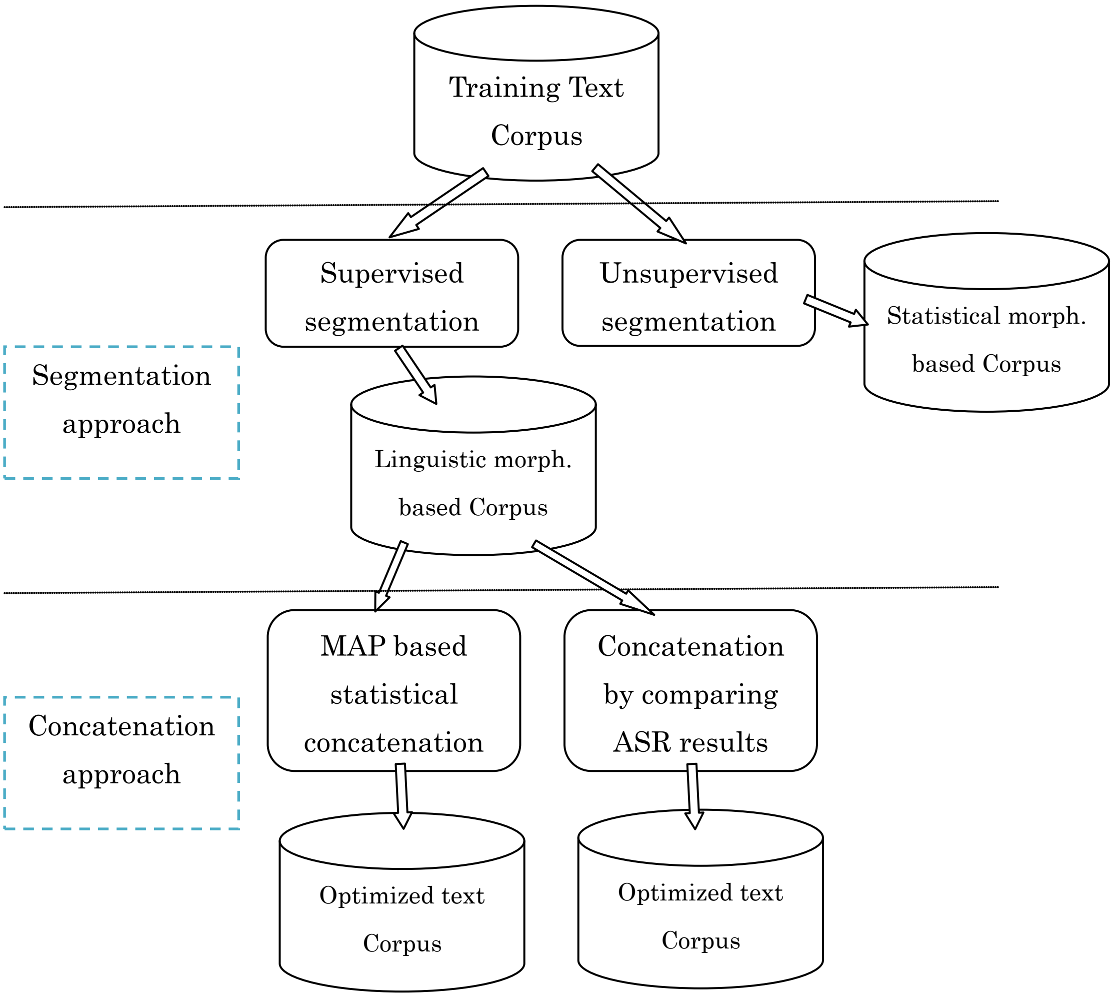

Training Text Corpus
Supervised segmentation
Unsupervised segmentation
Statistical morph. based Corpus
Segmentation approach
Linguistic morph. based Corpus
MAP based statistical concatenation
Concatenation by comparing ASR results
Concatenation approach
Optimized text Corpus
Optimized text Corpus

Figure 6.1 Pipeline of segmentation and concatenation approaches.

6.2 Comparison of segmentation approaches

Morpheme unit, adopted as the standard ASR unit for *agglutinative* languages, provides the smallest basic unit set for lexicon optimization [83][88]. Generally there are two morpheme segmentation methods: one is supervised, and the other is the unsupervised segmentation.

6.2.1 Supervised morpheme segmentation

Supervised morpheme segmentation method needs a manually segmented training corpus. The surface realizations of the morphological structure are constrained and modified by a number of language phenomenon such as insertion, deletion, phonetic harmony, and disharmony [107].

An Uyghur morpheme segmenter has been developed by using a statistical method, as shown in Section 3.3. We built a probabilistic model in a supervised way by training with the manually prepared training corpus. The word coverage is 86.85%. Morpheme coverage is 98.44%. The morpheme segmentation accuracy is **97.66%** which is the percentage of the exact match of all morphemes in automatic segmentation compared against manual segmentation.

6.2.2 Unsupervised morpheme segmentation

Unsupervised morpheme segmentation extracts *pseudo-morpheme* units based on a raw text corpus as discussed in section 2.3.2. Linguistic constrains like the phonetic harmony and morphological changes make the unsupervised segmentation difficult to compete with the supervised segmentation method. However, extraction of *linguistic morpheme*s is not the main target of all unsupervised methods [55-73]. The practical purpose of the segmentation is to provide a vocabulary which is smaller and generalizes better than a vocabulary consisting of words as they appear in text. Such a vocabulary could be utilized in statistical LM e.g., for ASR. As opposed to the supervised morpheme segmentation which needs manual labor separately for every language, the unsupervised method automatically discover morpheme-like units from the un-annotated text corpora, and applicable to any language with concatenative morphology.

Morfessor developed by Creutz [55] is a general model for the unsupervised induction of a simple morphology from a raw text data. Morfessor has been designed to cope with the languages having predominately a concatenative morphology and where the number of *morphs* per word can vary much and is not known in advance. It is based on the probabilistic maximum a posteriori (MAP) formulation. This main idea in this method is to use frequent word units to segment infrequent words. *Morfessor* model is successfully applied for ASR experiments for several languages such as Finnish and Turkish [58], and reported to have improved ASR performance compared to the word based model. We use the *Morfessor* tool to split words into *statistical morphemes*.

6.2.3 LM evaluation

The statistical properties of the lexicon size, coverage, perplexity based on n-gram LMs are compared for all the basic units as shown in Table 3.10 and Table 3.11. Here we used the same training and test corpus as in Section 3.3.

Then, we compare n-gram models with different n sizes. Because of the memory limitation, we can only calculate up to 5-grams. To compare the results, the perplexity is normalized in reference to the word unit. Table 3.12 shows the result.

The morpheme model is significantly improved with longer n-grams, and the *linguistic morpheme*-based model performs better in coverage and perplexity compared to other units. We can see from the statistics in Table 6.2-6.4 that the *statistical morpheme* units have the similar statistics with the word units, thus have limited ability of further improvements. Furthermore, morpheme-like units are not fixed as they are dependent on the training corpus.

Table 6.2 Statistics of trigram LMs on various units (word and morpheme units are consistent with Table 3.9).

training corpus (sentences)			77.5K	155K	310K	620K
subset of learning corpus L			1/8	1/4	1/2	1/1
Vocabulary	word	Freq>0	149347	222729	329370	480067
		Freq>1	63501	97461	149054	227101
	morph	Freq>0	40403	61146	93627	144765
		Freq>1	17823	25145	37202	57768
	statistical morph	Freq>0	119707	170363	248756	381533
		Freq>1	60450	82768	106146	130466
tokens	word		1,453,870	2,904,037	5,806,217	11,587,471
	morpheme		2,748,350	5,487,041	10,965,894	21,869,762
	statistical morph		1,500,930	2,997,840	5,994,135	11,961,354
unigram coverage (%)	word		91.47	93.71	95.47	96.71
	morpheme		98.76	99.02	99.25	99.40
	statistical morph		93.88	95.85	97.03	97.55
bigram coverage (%)	word		53.07	58.64	64.56	71.10
	morpheme		89.60	92.07	94.08	95.77
	statistical morph		52.81	59.17	66.15	73.63
trigram coverage (%)	word		16.93	21.07	26.79	34.88
	morpheme		64.48	70.32	75.82	81.14
	statistical morph		15.99	20.46	26.78	35.65
perplexity	word		12856	6857	3689	1929
	morpheme		91.27	77.43	66.20	56.99
	statistical morph		7206	3920	2245	1286
normalized perplexity by word	word		12856	6857	3689	1929
	morpheme		5078.4	3706.8	2747.6	2059.8
	statistical morph		9592	5116	2878	1620

Table 6.3 Perplexity by trigram models of various units (word and morpheme units are consistent with Table 3.10).

training corpus	perplexity			perplexity normalized by words	
	word	morpheme	*statistical morphs*	morpheme	*statistical morphs*
L 1/64	125554	178	76190	18137	109419
L 1/32	53295	136	31096	10813	43387
L 1/16	25196	110	14275	7263	19424
L 1/8	12856	91	7206	5078	9592
L 1/4	6858	77	3920	3707	5116
L 1/2	3689	66	2245	2447	2878
L 1/1	1929	57	1286	2060	1620

Table 6.4 Normalized perplexity of n-gram models of various units (word and morpheme units are consistent with Table 3.11).

unit	word	morph	statistical morph.
vocabulary size	227.1k	57.7k	132.8k
1-gram	29,302	482,973	29160
2-gram	3039	6294	2632
3-gram	1929	2060	1620
4-gram	1754	1318	1436
5-gram	1700	1091	1370

6.2.4 ASR evaluation

The LMs have been evaluated in the Uyghur LVCSR task described in Section 3.5. The training and test sets are same as shown in Section 3.5.

We examine ASR performance with various n-gram LMs. Because the morpheme-based model is benefited from a much smaller vocabulary size, 4-gram LM performs best for morphemes while 3-gram is best for word-based LM. The *cutoff* threshold affects the

ASR performance only for the word-based LM.

Table 6.5 ASR results for various segmentation units (consistent with Table 3.14).

Baseline models		WER (%)	lexicon size	OOV rate
word 3-gram	cutoff-2	25.72	227.9k	2.8%
	cutoff-5	26.64	108.1k	4.4%
morpheme 4-gram	cutoff-2	27.92	55.2k	0.3%
	cutoff-5	28.11	27.4k	0.7%
statistical *morphs* 4-gram	cutoff-5	25.01	94.5k	0.9%
	cutoff-2	25.04	133.4k	0.8%

The ASR performance is shown in Table 6.5. It is observed that the *statistical morpheme*-based LM outperforms the *linguistic morpheme*-based LM and also the word-based LM.

6.3 Comparison of concatenation approaches

In the previous chapters, the morpheme optimization methods based on comparing ASR results of word and morpheme based models are discussed in details. The extraction of problematic morpheme sequences can be implemented in both manual and automatic ways.

6.3.1 Effects of manually selected features

By comparing the ASR results of word and morpheme based results. We can manually extract problematic morpheme sequence according to some features. Below are some of the effective features.

(1) Error frequency: from the training data, the words misrecognized more than twice are extracted, and added to the vocabulary.

(2) Unit length: the morphemes consisting of single phoneme are merged to each other or to the previous morpheme.

(3) Stem & word-ending:

We confirm an accumulative effect of the features. The final result here outperforms the word-based model result with a much smaller vocabulary size. The ASR results are summarized in Table 6.6

Table 6.6 WER reduction based on the manually extracted features (same as Table 4.6).

Models	WER (%)	△WER (%)	lexicon size
Morpheme-based baseline	28.11	-	27.4k
Error frequency feature	26.11	2.00	40.4k
Length feature	27.19	0.92	32.9k
Attribute feature	26.74	1.36	36.3k
Attribute feature+ Length feature	25.80	2.31	41.3k
Attribute feature+ Length feature+ Error freq feature	24.89	3.22	56.7k

6.3.2 Machine learning of effective units

Although the manual extraction of problematic morpheme sequences is very simple, this method labels all units that have the relevant feature into problematic units, thus may cause an over-concatenation problem by unnecessarily concatenating morpheme sequences that works well in ASR. It is necessary to automate the process for optimization.

The discriminative approach to lexicon optimization method presented in the Chapter 5 can automatically sort out the problematic morpheme sequences by feeding the aligned ASR results of word and morpheme units into a machine learning algorithm such as SVM and LR. We found that the sub-word bigram feature and SVM can produce the best result. Table 6.7 shows the summary of results.

Table 6.7 ASR results for machine learning methods (same as Table 5.11).

Discriminative learning methods	WER (%)	lexicon size	OOV rate
Morpheme-based baseline	28.11	27.4k	0.7%
perceptron	25.27	49.9k	-
LR	24.87	65.8k	0.9%
SVM	24.61	55.1k	0.9%

6.3.3 Discriminative concatenation of statistical morphemes

The discriminative concatenation approach can also be applied to the *statistical morpheme*s. But the *statistical morpheme*s have a lower WER than the word-based model, so the basic assumption that the longer units have better ASR performance than the shorter units is no longer valid. Below Table 6.8 shows the results. No further improvements are demonstrated due to the lack of critical samples.

Table 6.8 Discriminative concatenation of *statistical morpheme*s.

models	WER (%)	lexicon size	OOV rate
word-based baseline	25.77	227.9k	2.8%
statistical morpheme-based	25.04	133.4k	0.8%
discriminative optimization for statistical morphemes (SVM, bigram feature)	25.11	140.5k	0.8%

6.3.4 Concatenation method based on statistical model

A large speech training database and its ASR results based on word and morpheme units are necessary for the optimization methods discussed in the previous sub-sections.

Here we explore an application of the probabilistic scheme applied to concatenation of *linguistic morpheme* sequences. In this study, we revise the Morfessor tool and use its maximum a posteriori (MAP) criterion for the concatenation of *linguistic morpheme*.

Table 6.9 ASR result of direct statistical concatenation optimization and comparison with baseline morpheme based LM.

4-gram models		WER (%)	lexicon size	OOV rate
Linguistic morpheme	Cutoff-5	28.11	27.4k	0.7%
	Cutoff-2	27.92	55.2k	0.3%
Statistical concatenation	Cutoff-5	24.96	98.35k	0.9%
	Cutoff-2	24.85	139.0k	0.8%

We can simply apply the tool to morpheme concatenation. As the input of this tool is the words with features and output is the optimized sub-word sequences (*statistical morpheme*s), we can simply insert morpheme boundary to the input words. Then, we confine the searching point of the *Morfessor* algorithm only to the morpheme boundaries rather than the character boundaries. In this case, the smallest unit here is the morpheme instead of the phoneme or character. As a result, the output is some kind of regrouping of morpheme sequences or in other words concatenation of *linguistic morpheme* units. The ASR results based on this unit are shown in Table 6.9

6.4 Summary of segmentation and concatenation approaches

We have investigated a variety of segmentation and concatenation approaches to lexicon optimization. The results are summarized in Table 6.10.

ASR results show that the *statistical morpheme*s outperform the *linguistic morphemes*. However, the lexicon size of *linguistic morpheme* is smaller compared to that of *statistical morpheme*. Therefore, our concatenation approach is based on the *linguistic morpheme* sequences. A variety of manual and automatic concatenation methods significantly reduced the WER with a small lexicon size compared to the word units. The proposed discriminative approach is also applied to the *statistical morpheme*s. But no additional gain is obtained, because the words unit no longer outperformed the *statistical morpheme*s therefore cannot provide enough critical samples. Among the various lexicon optimization methods, the proposed discriminative approach produced the best ASR result in terms of WER and lexicon size. Actually, only this method achieved a statistically significant improvement from the word-based model.

Table 6.10 Summary of ASR results based on segmentation and concatenation approaches.

approaches		WER (%)	lexicon size	OOV rate
Word-based baseline best result		25.72	227.9k	2.8%
segmentation	Linguistic morpheme (Chapter 3; baseline)	27.92	55.2k	0.3%
	Unsupervised *morphs* (Morfessor)	25.01	94.5k	0.9%
concatenation	Manual extraction of features (Chapter 4)	24.89	56.7k	-
	Machine learning approach (SVM) (Chapter 5)	24.61	55.1k	0.9%
	Statistical concatenation approach (Section 6.3.4)	24.96	98.4k	0.9%

Chapter 7

Concluding Remarks

7.1 Summary of the work

This thesis has addressed the fundamental problems related with the selection of basic lexicon unit for ASR of *agglutinative* languages. Several lexicon optimization methods are investigated, before proposing the discriminative approach based on comparison the two layers of ASR results.

Selection of lexicon unit is an important issue for *agglutinative* languages, word units have a variety of derivatives and the vocabulary size increases explosively with the corpus size. As the morpheme units can provide a good coverage and better linguistic properties, we selected the morpheme unit, the smallest meaning bearing unit, as the basis of our concatenative lexicon optimization. The predefined *linguistic morphemes* may not fit the statistical modeling of ASR system, and there are the phonological and morphological changes in the surface forms when we split words into smaller units. But the morphemes preserve linguistic information as well as word boundary, thus can be an ideal basic lexicon set for concatenation approaches.

Several data-driven concatenation approaches to lexicon optimization are investigated. Most of the existing statistical methods are based on formulations which are splitting infrequent and longer units to reduce OOV while concatenating frequently co-occurred units to reduce confusion. However, they are not actually formulated in terms of direct reduction of WER in ASR.

Therefore, we propose a novel discriminative approach which is based on comparison of the two layers of ASR results. Specifically, we implement it with Support Vector Machines (SVM) and Logistic Regression (LR) model as well as simple perceptron. The SVM and LR are more robustly trained, and SVM results in the best performance with a large dimension of features. The proposed learning scheme is realized in an unsupervised manner in that it does not need correct transcription for training data.

The optimized method is very stable, and can reduce WER and the lexicon size significantly. The higher cutoff rates do not increase the WER while still significantly reducing the lexicon size. The proposed method is least affected by the changes of training parameters.

Based on the proposed method, we have developed an LVCSR system for Uyghur language for the first time with a high accuracy. As an under-resourced language, it is difficult to collect speech and text corpora, and more difficult to prepare annotated training corpora. In this thesis, we have summarized the standard structures of morphological units of Uyghur language, and presented the meaningful results on various unit sets.

7.2 Discussion on generality of the proposed approach

The proposed discriminative lexicon optimization approach is implemented in an under-resourced language. At this moment, we have implemented and evaluated only for the Uyghur language in this work. The generality of the proposed method is discussed in this final section.

The proposed method is based on the comparison of ASR results of word (a longer unit set) and morpheme (a shorter unit set) based systems. The basic assumption is the word unit based ASR system has a lower WER than those of the morpheme unit. This assumption has already been demonstrated in many inflectional languages [82-103]. Work on the Turkish language, which has the same morphological structure with Uyghur, reported that the morpheme units are easily confused in ASR compared to the word units [87][88]. Similar tendency is also reported in Korean [82-84]. Thus, concatenation approaches are reported to have improved ASR performance in these languages.

We have a difficulty in implementing the approach in other languages, as a morphological analyzer is necessary to segment text to words and also *linguistic morphemes*. For the Japanese language, an *agglutinative* language, it is necessary to segment text to words or morphemes as there are no word or morpheme delimiters. But there is no corpus or segmentation system that segments text into both word and morpheme units consistently. Furthermore, with a large amount of training data, the longer unit based model could have a better coverage and can catch longer context, especially for resource-rich languages such as Japanese and Chinese.

In the n-gram language modeling, a larger training corpus favors the longer unit set with smaller perplexity and enough coverage, and therefore better ASR performance. However, we believe that the proposed discriminative approach is useful for a number of *agglutinative* languages in the world, that do not have a large population and a large corpus.

Bibliography

[1] P.Brown, P.deSouza, R.Mercer, V.D.Piera, J.Lai, “Class-based n-gram models of natural language,” Computational linguistics, 1992.

[2] P. F. Brown, Vincent J. Della Pietra, Robert L. Mercer, Stephen A. Della Pietra, Jennifer C. Lai , “An Estimate of an Upper Bound for the Entropy of English”, Computational Linguistics, 1992.

[3] K. Hwang, “Vocabulary optimization based on perplexity,” ICASSP 1997.

[4] T. Shinozaki, S. Furui, “A new lexicon optimization method for LVCSR based on linguistic and acoustic characteristics of words,” Interspeech 2002.

[5] R. Kneser and H. Ney, “Improved backing-off for m-gram language modeling,” Proceedings of the ICASSP, pages 181–184, 1995.

[6] G. Saon, M. Padmanabhan, “Data-Driven Approach to Designing Compound Words for Continuous Speech recognition,” IEEE Trans. Speech and Audio Processing, Vol.9, No.4, 2001.

[7] T. Kawahara et al., “Free software toolkit for Japanese large vocabulary continuous speech recognition,” In Proc. ICSLP, Vol.4, pp.476–479, 2000.

[8] T. Kawahara et al., “Recent progress of open source LVCSR engine Julius and Japanese model repository,” In Proc. ISCA, pp.3069–3072, 2004.

[9] E. Whittaker, P. Woodland, “Particle-based language modeling,” Proc. ISCSLP, Beijing, China, Oct. 2003.

[10] M. Ostendorf, I. Shafran, R. Bates, “Prosody Models for Conversational Speech Recognition,” Proc. of the 2nd Plenary Meeting and Symposium on Prosody and Speech Processing, 2003.

[11] Jun Zhao, “lexicon optimization for Chinese language modeling,” Proc. Of International Symposium Conference on Spoken Language Processing, 2000.

[12] M. Adda-Decker, L. Lamel, “The use of lexica in automatic speech recognition,” Lexicon Development for Speech and Language Processing, pages 43–75, Kluwer Academic Publishers. Dordrecht. (2000).

[13] J. L. Hieronymus, D. McKelvie, F. R. McInnes, “Use of acoustic sentence level and lexical stress in HSMM speech recognition,” Proc. ICASSP 1992, pages I−225−227.

[14] M. Ablimit, G. Neubig, M. Mimura, S. Mori, T. Kawahara, A. Hamdulla, “Uyghur Morpheme-based Language Models and ASR,” In Proc. IEEE-ICSP, 2010.

[15] M. Ablimit, M. Eli, and T. Kawahara, "Partly-Supervised Uyghur morpheme segmentation," In Proc. Oriental-COCOSDA Workshop, pp.71–76, 2008.

[16] M. Ablimit, T. Kawahara, A. Hamdulla, "Discriminative approach to lexical entry selection for Automatic Speech Recognition of *agglutinative* language," In Proc. ICASSP 2012.

[17] M. Ablimit, A. Hamdulla, T. Kawahara, "Morpheme Concatenation Approach in Language Modeling for Large-Vocabulary Uyghur Speech Recognition," In Proc. Oriental-COCOSDA Workshop, 2011.

[18] M. Ablimit, T. Kawahara, A. Hamdulla, "Lexicon Optimization for Automatic Speech Recognition based on Discriminative Learning," In Proc. APSIPA, 2011.

[19] M. Ablimit, T. Kawahara, A. Hamdulla, "Discriminative approach to lexical entry selection for Automatic Speech Recognition of *agglutinative* language," In Proc. ICASSP 2012.

[20] M. Ablimit, T. Kawahara, A. Hamdulla, "Lexicon Optimization based on Discriminative Learning for Automatic Speech Recognition of *Agglutinative* Language," Speech Communication, (under review).

[21] Batuer Aisha, Maosong Sun, "A Statistical Method for Uyghur Tokenization," Proceedings of IEEE International Conference on Natural Language. 2009.

[22] Askar Hamdulla, Dilmurat Tursun "An Acoustic Parametric Database for Uyghur Language," Proceedings of the 2009 International Joint Conference on Artificial Intelligence.

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," Journal of Machine Learning Research 9(2008), 1871-1874.

[24] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin, "Dual Coordinate Descent Methods for Logistic Regression and Maximum Entropy Models," Machine Learning, 85(2011), 41-75.

[25] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. Sundararajan, and S. Sathiya Keerthi, "A Dual Coordinate Descent Method for Large-scale Linear SVM," ICML 2008.

[26] Christopher, J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Neural Computation, 1299-1319, 1998.

[27] Hong cui, Wang, "A flexible computer assisted language learning system with speech recognition and error detection capability," PhD thesis, Kyoto University, 2010.

[28] J. Bellegarda, "Exploiting latent semantic information in statistical language modeling," Proceedings of the IEEE, 88(8):1279–1296, 2000.

[29] Brian Roark, Murat Saraclar, Michael Collins, Mark Johnson, "Discriminative Language Modeling with Conditional Random Fields and the Perceptron Algorithm," In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2004.

[30] Brian Roark, Murat Saraclar, and Michael Collins, "Discriminative n-gram language modeling," Computer Speech and Language, 21(2):373–392, 2007.

[31] E. Arisoy, M. Saraclar, B. Roark, I. Shafran, "Discriminative language modeling with linguistic and statistically derived features," IEEE TASLP 2012

[32] M. Collins, B. Roark, M. Saraclar, "Discriminative syntactic language modeling for speech recognition," In Proc. ACL, pages 507–514, 2005.

[33] Michiel Bacchiani, Brian Roark and Murat Saraçlar, "Language model adaptation with MAP estimation and the perceptron algorithm," In Proceedings of the Human Language Technology Conference and Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pages 21-24. 2004.

[34] M. Collins, "Discriminative training methods for HMMs: Theory and experiments with perceptron algorithm," In Proc. EMNLP 2002.

[35] Hong-Kwang Jeff Kuo, Eric Fosler-Lussier, Hui Jiang, Chin-Hui Lee, "Discriminative training of language models for speech recognition," ICASSP 2002: 325-328

[36] Zheng Chen, Kai-Fu Lee, Ming-jing Li "Discriminative Training on Language Model," in Proc. ICSLP, 2000.

[37] Michael R. Brent, "An efficient, probabilistically sound algorithm for segmentation and word discovery," Machine Learning, 34:71–105, 1999.

[38] F. Jelinek, R. Mercer, S. Roukous, "Classifying words for improved statistical language models," ICASSP 1990.

[39] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," In International Conference on Machine Learning (ICML), 2001.

[40] S. Novotney, R. Schwartz, J. Ma, "Unsupervised acoustic and language model training with small amounts of labeled data," ICASSP 2009.

[41] H. Poon, C. Cherry, K. Toutanova, "Unsupervised morphological segmentation with log-linear models," In NAACL, pages 209–217, 2009.

[42] Carl de Marcken, "The unsupervised acquisition of a lexicon from continuous speech," Technical Report A.I. Memo, MIT Artificial Intelligence Lab., Cambridge, Massachusetts, 1995.

[43] Carl de Marcken, “Linguistic structure as composition and perturbation,” In Meeting of the Association for Computational Linguistics, 1996.

[44] Herv´e D´ejean, “Morphemes as necessary concept for structures discovery from untagged corpora,” In Workshop on Paradigms and Grounding in Natural Language Learning, pages 295–299, Adelaide, Jan.22, 1998

[45] Sabine Deligne and Fr´ed´eric Bimbot, “Inference of variable-length linguistic and acoustic units by multi-grams,” Speech Communication, 23:223–241, 1997.

[46] John Goldsmith, “Unsupervised learning of the morphology of a natural language,” Computational Linguistics, 27(2):153–198. 2001.

[47] Yu Hua, “Unsupervised word induction using MDL criterion,” In Proceedings of ISCSL, Beijing, 2000.

[48] Patrick Schone, Daniel Jurafsky, “Knowledge free induction of morphology using latent semantic analysis,” In Proceedings of CoNLL-2000 and LLL-2000. Lisbon

[49] R. Harald Baayen, “Word Frequency Distributions,” Kluwer Academic Publishers, 2001.

[50] Stanley F. Chen, “Building Probabilistic Models for Natural Language,” PhD. thesis, Harvard University, 1996.

[51] Mathias Creutz and Krista Lagus, “Unsupervised discovery of morphemes,” In Proc. Workshop on Morphological and Phonological Learning of ACL’02, pages 21–30, Philadelphia, Pennsylvania, USA, 2002.

[52] Masaaki Nagata, “A self-organizing Japanese word segmenter using heuristic word identification and re-estimation,” In Proc. Fifth workshop on very large corpora, pages 203–215. 1997..

[53] V. Siivola, T. Hirsim¨aki, M. Creutz, M. Kurimo, “Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner,” In Proc. Eurospeech 2003, pages 2293–2296, Geneva, Switzerland, 2003.

[54] John Goldsmith, “Unsupervised learning of the morphology of a natural language”, Computational linguistics, vol. 27, June. 2001

[55] M. Creutz, “Introduction of the morphology of natural language: Unsupervised morpheme segmentation with application to automatic speech recognition”, PhD. Dissertation, Helsinki University of Technology, Finland 2006.

[56] M. Creutz, K. Legus, “Unsupervised discovery of morphemes,” In producing of the ACL-02 workshop on morphological and phonological science, 2005.

[57] M. Creutz, K. Legus, “Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0,” Publication in Computer and

information science, Report A81, Helsinki University of Technology, March, 2005.

[58] M. Creutz, T. Hirsimaki, M. Kurimo, A. Puurula, J. Pylkkonen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraclar, and A. Stolcke. “Morph-based speech recognition and the modeling of out-of-vocabulary words across languages,” ACM transactions, Speech Lang. Process., vol. 5, no. 1, pp. 1-29, 2007.

[59] Maria Carolina Parada, “Learning sub-word units and exploiting contextual information for open vocabulary speech recognition”, PhD dissertation thesis, Johns Hopkins University, 2011.

[60] H. Aronowitz, “Online vocabulary adaptation using contextual information and information retrieval,” Interspeech, 2008.

[61] A. Asadi, R. Schwartz, and J. Makhoul, “Automatic detection of new words in a large vocabulary continuous speech recognition system,” ICASSP, pages 125–128, 1989.

[62] L. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains,” Annals of Mathematical Statistics, 37:1559–1563, 1966.

[63] I. Bazzi, “Modeling Out-of-Vocabulary Words for Robust Speech Recognition,” PhD thesis, Massachusetts Institute of Technology, 2002.

[64] I. Bazzi and J. Glass, “Heterogeneous lexical units for automatic speech recognition: Preliminary investigations,” ICASSP, pages 1257–1260, 2000.

[65] I. Bazzi and J. Glass “Learning units for domain-independent out-of-vocabulary word modeling,” In Eurospeech, 2001.

[66] G. Choueiter, “Linguistically-motivated sub-word modeling with applications to speech recognition,” PhD thesis, Massachusetts Institute of Technology, 2009.

[67] B. Decadt, J. Duchateau, W. Daelemans, P. Wambacq, “Transcription of out-of-vocabulary words in large vocabulary speech recognition based on phoneme-to-grapheme conversion,” ICASSP 2002.

[68] L. Galescu, “Recognition of out-of-vocabulary words with sub-lexical language models,” Eurospeech, pages 249–252, 2003.

[69] Haeb-Umbach, Ney, “Linear discriminant analysis for improved large vocabulary continuous speech recognition,” ICASSP, 1:13–16, 1992.

[70] T. J. Hazen, I. Bazzi, “A comparison and combination of methods for OOV word detection and word confidence scoring,” In Proceedings of the International Conference on Acoustics, 2001.

[71] A. Rastrow, A. Sethy, B. Ramabhadran, F. Jelinek, “Towards using hybrid, word, and fragment units for vocabulary independent LVCSR systems,” Interspeech, 2009.

[72] Graham Neubig, “Unsupervised Learning of Lexical Information for Language Processing Systems,” PhD thesis, Kyoto University. 2012.

[73] Graham Neubig, Masato Mimura, Shinsuke Mori, Tatsuya Kawahara, “Bayesian Learning of a Language Model from Continuous Speech,” IEICE Transactions on Information and Systems (link), E95-D-2, pp.614-625. February 2012.

[74] G. Neubig, M. Mimura, S. Mori, T. Kawahara, “Bayesian learning of a language model from continuous speech,” IEICE transactions , 2012.

[75] Mohsen Arabsorkhi, Mehrnoush Shamsfard, “Unsupervised discovery of Persian Morphemes,” 11th International CSI Computer Conference, Tehran, Feb. 2006

[76] Stefan Bordag, “Unsupervised and knowledge-free morpheme segmentation and analysis,” Morphochallenge, 2007

[77] Thomas Pellegrini, Lori Lamel, “Using phonetic features in unsupervised word decompounding for ASR with application to a less-represented language,” Interspeech, 2007.

[78] Thomas Pellegrini, Lori Lamel, “Using phonetic features in unsupervised word decompounding for ASR with application to a less-represented language,” Interspeech, 2007.

[79] Kemal Oflazer, “Two-level Description of Turkish Morphology,” Literary and Linguistic Computing, 1994.

[80] Richard Sproat, “Book Reviews PC-KIMMO: A Two-Level Processor for Morphological Analysis,” AT& T Bell Laboratories, 1990.

[81] D. Kiecza, T. Schultz, A. Waibel, “Data-driven determination of appropriate dictionary units for Korean LVCSR,” Proc. ICSP Seoul, 1999.

[82] O.-W. Kwon and J. Park, “Korean large vocabulary continuous speech recognition with morpheme-based recognition units,” Speech Communication, vol. 39, pp. 287–300, 2003.

[83] O-W. Kwon, “Performance of LVCSR with morpheme-based and syllable-based recognition units,” In Proc. ICASSP, pp.1567–1570, 2000.

[84] O.-W. Kwon, K. Hwang, J. Park, “Korean LVCSR using pseudo-morpheme units,” Eurospeech, 1999.

[85] Hong-Kwang Jeff Kuo, Wolfgang Reichl, “Phrase-based language models for speech recognition,” Eurospeech, 1999.

[86] L. Galescu, “Recognition of out-of-vocabulary words with sub-lexical language models,” Eurospeech, pages 249–252, 2003.

[87] Çarkı, K., P. Geutner, and T. Schultz, “Turkish LVCSR: Towards Better Speech Recognition for *Agglutinative* Languages,” ICASSP--2000, Istanbul, Turkey, June.

[88] K. Hacioglu, B. Pellom, T. Ciloglu, O. Ozturk, M. Kurimo, M. Creutz. “On Lexicon Creation for Turkish LVCSR,” In Proc. Eurospeech, 2003.

[89] Ebru Arisoy, Hasim Sak, Murat Saraclar. “Language Modeling for Automatic Turkish Broadcast News Transcription,” In Proc. Interspeech, 2007.

[90] Roeland Ordelman, Arjan van Hessen, and Franciska de Jong, “Compound decomposition in Dutch large vocabulary speech recognition,” Eurospeech, Geneva, 2003.

[91] Martha Larson, Daniel Willett, Joachim Kohler, Gerhard Rigoll, “Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches,” Interspeech, 2000.

[92] Andre Berton, Pablo Fetter, Peter Regel-Brietzmann, “Compound Words in Large Vocabulary German Speech Recognition Systems,” In Proc. Int. Conf on Spoken Language Processing, Philadelphia, Australia, Sept. 1996.

[93] P. Geutner, “Using morphology towards better large-vocabulary speech recognition systems,” ICASSP, 1995.

[94] Markus Nuβbaum-Thom, Amr El-Desoky Mousa, Ralf Schluter, Hermann Ney, “Compound Word Recombination for German LVCSR,” Interpeech, 2011.

[95] A. El-Desoky, M. Shaik, R. Schl¨uter, and H. Ney, “Morpheme Based Factored Language Models for German LVCSR,” in Interspeech, 2012.

[96] B. Xiang, K. Nguyen, L. Nguyen, R. Schwartz, J. Makhoul, “Morphological decomposition for Arabic broadcast news transcription,” ICASSP 2006, Toulouse, France 2006.

[97] Dimitra Vergyri, Katrin Kirchhoff, Kevin Duh, Andreas Stolcke, “Morphology-Based Language Modeling for Arabic Speech Recognition,” Interspeech, 2004.

[98] M. Afify, R. Sarikaya, H.-K. J. Kuo, L. Besacier, and Y.Gao, “On the use of morphological analysis for dialectal Arabic speech recognition,” Interspeech, Pittsburgh, PA, USA, Sep. 2006.

[99] Amr El-Desoky, Christian Gollan, David Rybach, Ralf Schluter, Hermann Ney, “Investigating the use of morphological decomposition and diacrit,” Interspeech, 2009.

[100] Ruhi Sarikaya, Mohamed Afify, Yonggang Deng, Hakan Erdogan, Yuqing Gao, "Joint Morphological-Lexical Language Modeling for Processing Morphologically Rich Language with Application to Dialectal Arabic," IEEE Transactions on Audio, Speech & Language Processing - TASLP , vol. 16, no. 7, pp. 1330-1339, 2008.

[101] A. Puurula, "Vocabulary decomposition for Estonian open vocabulary speech recognition," In Proceedings of ACL, 2007.

[102] W. Byrne, J. Hajic, P. Jelinek, etc. "On large vocabulary continuous speech recognition of highly inflectional language – Czech," Eurospeech 2001, Aalborg, Denmark.

[103] M. Jongtaveesataporn, I. Thienlikit, C. Wutiwiwatchai, S. Furui, "Lexical units for Thai LVCSR," Speech Communication, pp.379–389, 2009

[104] Hakan Erdogan, Osman Buyuk, Kemal Oflazer, "Incorporating language constraints in sub-word based speech recognition," in Proceedings of ASRU 2005.

[105] Amr El-Desoky Mousa, Ralf Schluter, Hermann Ney, "Investigations on the use of morpheme level features in language models for Arabic LVCSR," ICASSP 2012.

[106] Christopher M. Bishop, "Pattern recognition and machine learning," Springer, 2006.

[107] "Uyghur spelling and pronunciation dictionary," Xinjiang people's publishing House, Urumqi, China, 1997.

[108] Sharon Goldwater, Thomas L. Griffiths, Mark Johnson, "A Bayesian framework for word segmentation: Exploring the effects of context," Cognition 112 (1), 21-54, 2009.

[109] G. Lidstone, "Notes on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities," Transactions of the Faculty of Actuaries, 8:182–192, 1920.

[110] F. Jelinek, "Statistical Methods for Speech Recognition," MIT Press, 1997.

[111] F. Jelinek, R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," In Proceedings of the Workshop on Pattern Recognition in Practice, Amsterdam, Netherlands, 1980.

[112] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-35(3):400–401, 1987.

[113] I. Witten, T. Bell, “The zero-frequency problem: Estimation the probability of novel events in adaptive text compression,” IEEE Transactions on information theory, 37(4): 1085–1094, 1991.

[114] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, pages 310–318, 1998.

[115] C. Kit, and Y. Wilks, “Unsupervised Learning of Word Boundary with Description Length Gain,” In Proceedings CoNLL99 ACL Workshop, Bergen, 1999

[116] M. Brent, T. Cartwright, “Distributional regularity and phonotactics are useful for segmentation,” Cognition 61, 93-125, 1996.

[117] Satanjeev Banerjee, Jack Mostow, Joseph Beck, and Wilson Tam, “Improving language models by learning from speech recognition errors in a reading tutor that listens,” In Proceedings of the Second International Conference on Applied Artificial Intelligence, Fort Panhala, Kolhapur, India. 2003.

[118] Laura Mayfield Tomokiyo and Klaus Ries, “An automatic method for learning a Japanese lexicon for recognition of spontaneous speech,” In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1998.

[119] Rie Kubota Ando and Lillian Lee, “Mostly unsupervised statistical segmentation of Japanese Application to Kanji,” Lee Proceedings of NAACL, pp. 241-- 248, 2000.

[120] Daichi Mochihashi, Takeshi Yamada, Naonori Ueda, “Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling,” Proc. 47th Annual meeting of the Association for Computational Linguistics, 2009.

List of Author's Publications

Refereed Papers

[1] M. Ablimit, M. Eli, and T. Kawahara, "Partly-Supervised Uyghur morpheme segmentation," In Proc. Oriental-COCOSDA Workshop, pp.71–76, 2008.

[2] M. Ablimit, G. Neubig, M. Mimura, S. Mori, T. Kawahara, A. Hamdulla, "Uyghur *Morpheme-based* Language Models and *ASR,"* In Proc. IEEE-ICSP, 2010.

[3] M. Ablimit, A. Hamdulla, T. Kawahara, "Morpheme Concatenation Approach in Language Modeling for Large-Vocabulary Uyghur Speech Recognition," In Proc. Oriental-COCOSDA Workshop, 2011.

[4] M. Ablimit, T. Kawahara, A. Hamdulla, "Lexicon Optimization for Automatic Speech Recognition based on Discriminative Learning," In Proc. APSIPA, 2011.

[5] M. Ablimit, T. Kawahara, A. Hamdulla, "Discriminative approach to lexical entry selection for Automatic Speech Recognition of *agglutinative* language," In Proc. ICASSP 2012.

[6] M. Ablimit, T. Kawahara, A. Hamdulla, "Lexicon Optimization based on Discriminative Learning for Automatic Speech Recognition of *Agglutinative* Language," Speech Communication, (under review).

Technical Reports

[1] M. Ablimit, G. Neubig, M. Mimura, S. Mori, T. Kawahara, A. Hamdulla, "Uyghur *Morpheme-based* Language Models and ASR," IPSJ Technical report, SLP-82-17 2010.

[2] M. Ablimit, T. Kawahara, A. Hamdulla, "Evaluation of Lexicon Optimization based on Discriminative Learning," IPSJ Technical report, SLP-89-2 2011.

[3] M. Ablimit, T. Kawahara, A. Hamdulla, "Comparison of Discriminative Models for Lexicon Optimization for ASR of *Agglutinative* Language," IPSJ Technical report, SLP-92-13 2012.