Morphology-Based Language Modeling for Amharic

Dissertationsschrift zum Erlangung des Grades eines Doktors der Naturwissenschaften am Department Informatik an der Fakultät für Mathematik, Informatik und Naturwissenschaften der Universität Hamburg

Vorgelegt von
Martha Yifiru Tachbelie
aus Addis Ababa, Äthiopia

August 2010

Betreut von:
Prof. Dr. Ing. Wolfgang Menzel, Universität Hamburg

Genehmigt von der Fakultät für Mathematik, Informatik und Naturwissenschaften Department Informatik der Universität Hamburg auf Antrag von

Prof. Dr. Wolfgang Menzel (Erstgutachter/Betreuer)
Prof. Dr. Christopher Habel (Zweitgutachter)
Prof. Dr. Leonie Dreschler-Fischer (Vorsitzender)

Hamburg, den 31.08.2010 (Tag der Disputation)

Dedicated to
my husband Solomon Teferra Abate
and
my daughter Deborah Solomon

Acknowledgment

First and for most, I would like to express my heartfelt thanks to **God**, who gave me the strength, determination, endurance and wisdom to bring this thesis to completion. Oh God, I would be nowhere without you. Your name be glorified.

I owe a great debt of gratitude to my supervisor **Prof. Dr. Ing. Wolfgang Menzel**. This thesis would not have been possible without your guidance, as well as inspiring and enlightening ideas, comments and suggestions. I am glad to be your supervisee and share your rich, broad and deep knowledge. Thank you very much. Thank you also for your concern and readiness to help me whenever I needed. You made me comfortable to come to you whenever I have problems. You facilitated favorable conditions for me to fulfill the course work I was required to do and spent your precious time in selecting relevant courses and text books, as well as in assessing me through long discussions. Thank you so much.

My heartfelt thanks also goes to **Prof. Dr. Christopher Habel**. Thank you very much for your constructive comments and suggestions on the draft of the thesis. Thank you also for arranging favorable conditions for me to fulfill the courses that I should take to have a better understanding of the area. You spent your precious time in selecting text books and in the assessment discussions. You were ready to help me whenever I knocked at your door. Thank you very much.

I am grateful to **Prof. Baye Yimam** for sharing his knowledge on Amharic morphology. You shared me your deep linguistic knowledge both through personal discussion and your wonderful book which I received as a gift from you. Many thanks.

Let me use this opportunity to express my deepest gratitude to my husband and colleague, **Dr. Solomon Teferra**, who is the source of my success in my professional career. Thank you for your love, care and patience. You sacrificed your professional career and let me finish my study. You replaced my role as a mother to our daughter. Soul, how can I ever express my thanks to you? Thank you very much also for your invaluable comments and suggestions throughout my study and on the draft of this thesis.

Dr. Kerstin Fischer deserves my heartfelt gratitude for being besides me when I was in trouble. You gave me hope when I was almost to quit my study due to financial problems. Kerstin, this thesis would not have been possible, at least at this time, without your financial support. You facilitated the completion of the thesis. Many thanks for your kindness.

I would also like to thank all my colleagues at NATs for being friendly and making the atmosphere nice. Specifically, I would like to thank **Monica Gavrila**

for being with me in all times. Thank you for everything you have done for me and for encouraging me whenever I am down due to various reasons. Thank you very much also for printing, binding and submitting the thesis.

Yared Getachew and **Eleni Asfaw**, thank you so much for expressing the love of Jesus Christ to me. Thank you very much for everything you did for me and, above all, for your prayer support.

I am grateful to my mother, **W/o Asrat Demeke**, for her love and encouragement, as well as for bearing with my stay abroad. Thank you very much, Asratie. My love and thanks goes to my sister, **Eden (Mitin)**, for bearing with my stay abroad. Mitinie, I do not forget your usual question, "when do you come back to Ethiopia", whenever we spoke over the phone.

My love and special thanks goes to my daughter, **Deborah**, for being patient to stay most of the time without me. Many times, I arrive home after you went to bed. Thank you for your patience.

Last, but not least, I would like to thank the **University of Hamburg** for the financial support without which the study would not have been possible.

Martha Yifiru Tachbelie

August, 2010

Abstract

Language models are fundamental for many natural language processing applications. The most widely used type of language models are the corpus-based probabilistic ones. These models provide an estimate of the probability of a word sequence W based on training data. Therefore, large amounts of training data are required in order to ensure statistical significance. But even if the training data are very large, it is impossible to avoid the problems of data sparseness and out-of-vocabulary (OOV) words. These problems are particularly serious for languages with a rich morphology, which are characterized with high vocabulary growth rate and a correspondingly high perplexity of their language models. Since the vocabulary size directly affects system complexity, a promising direction is towards the use of sub-word units in language modeling.

This study explored different ways of language modeling for Amharic, a morphologically rich Semitic language, using morphemes as units. Morpheme-based language models have been trained on automatically and manually segmented data using the SRI Language Modeling toolkit (SRILM). The quality of these models has been assessed in terms of perplexity, the probability they assign to the test set, and the improvement in word recognition accuracy obtained as a result of using them in a speech recognition task. The results show that the morpheme-based language models trained on manually segmented data always have a higher quality.

A comparison with word-based models reveals that the word-based models fared better in terms of the probability they assigned to the test set. In terms of word recognition accuracy, however, interpolated (morpheme- and word-based) models achieved the best results. In addition, the morpheme-based models reduced the OOV rate considerably.

Since using morpheme-based language models in a lattice rescoring framework does not solve the OOV problem, speech recognition experiments in which morphemes are used as dictionary entries and language modeling units have been conducted. The use of morphemes highly reduced the OOV rate and consequently boosted the word recognition accuracy of the 5k vocabulary morpheme-based speech recognition system. However, as morpheme-based recognition systems suffer from acoustic confusability and limited n-gram language model scope, their performance with a larger morph vocabulary was not as expected.

When morphemes are used as units in language modeling, word-level dependencies might be lost. As a solution to this problem we have investigated root-based language models in the framework of factored language modeling. Although this produced far better test set probabilities, the much weaker predictions of a root-

based model resulted in a loss in word recognition accuracy. In addition to the morpheme-based language models, several factored language models that integrate morphological information into word based models have also been developed. The results show that integrating morphological information leads to better models.

In summary, the study showed that using morphemes in modeling morphologically rich languages is advantageous, especially in reducing the OOV rate. This, consequently, improves word recognition accuracy of small vocabulary morpheme-based speech recognition systems. Moreover, using morpheme-based language models as a complementary tool to the word-based models is fruitful. The study has also confirmed that the best way of evaluating a language model is by applying it to the application for which it was intended. Currently, this is the only way to reliably determine the actual contribution of the model to the performance of the target application.

Zusamenfassung

Sprachmodelle sind eine wichtige Grundlagen für viele Anwendungen der Verarbeitung natürlicher Sprache. Am weitesten verbreitet sind hierbei die korpusbasierten probabilistischen Ansätze. Diese Modelle erlauben es, die Wahrscheinlichkeit einer Wortsequenz W auf der Grundlage von Trainingsdaten abzuschätzen. Dazu werden große Mengen an Trainingsdaten benötigt, um die statistische Signifikanz sicherzustellen. Allerdings lässt sich auch beim Vorhandensein sehr großer Datenmengen das Problem der Datenknappheit und der lexikalischen Unvollständigkeit (out-of-vocabulary, OOV) nicht vollständig vermeiden.

Diese Probleme sind besonders gravierend für Sprachen mit einer reichhaltigen Morphologie, in denen der Wortschatz stark anwächst und zu Sprachmodellen mit hoher Perplexität führt. Da die Größe des Lexikons nicht beliebig gesteigert werden kann, besteht ein vielversprechender Ansatz in der Verwendung von Wortbestandteilen.

Diese Arbeit hat sich das Ziel gestellt, einen optimalen Weg zur Modellierung der amharischen Sprache, einer morphologisch reichhaltigen semitischen Sprache, zu finden, wobei als Wortuntereinheiten Morpheme verwendet werden. Mit Hilfe des SRI Language Modeling toolkit (SRILM) wurde eine Reihe morphem-basierter Sprachmodelle sowohl auf automatisch als auch auf manuell segmentierten Korpusdaten trainiert. Der Vergleich dieser Modelle erfolgt hinsichtlich ihrer Perplexität, der für eine Testdatenmenge geschätzten Wahrscheinlichkeit, sowie der Steigerung der Worterkennungsrate, die sich durch ihre Verwendung in einem Spracherkenner erzielen lässt.

Die automatisierte Ermittlung der Wortsegmentierung erfolgt mit Hilfe einer unüberwacht trainierten, korpus-basierten morphologischen Analyse (Morfessor). Die Resultate zeigen jedoch, dass die Sprachmodelle auf der Basis manuell segmentierter Daten generell besser sind.

Die morphem-basierten Sprachmodelle wurden auch mit wort-basierten Modellen verglichen. Dabei zeigt sich, dass die wort-basierten Modelle hinsichtlich der Testdatenwahrscheinlichkeit besser abschneiden. Im Hinblick auf die Gemauigkeit der Spracherkennung ergeben jedoch interpolierte (morphem- und wort-basierten) Modelle die besten Ergebnisse durch Neubewertung von Worthypothesegraphen. Darüberhinaus waren die morphem-basierten Models in der Lage, die OOV-Rate drastisch zu reduzieren.

Da sich durch die Verwendung morphem-basierter Sprachmodelle zur Neubewertung von Worthypothesegraphen das OOV-Problem nicht lösen lässt, wurden auch Experimente mit einem Spracherkennungssystem durchgeführt, das die Morpheme

direkt als Wörterbucheintragung und als Basiseinheit zur Sprachmodellierung verwendet. Dabei wurde durch die Verwendung der Morphemes die OOV-Rate erheblich reduziert und die Morphemerkennungsrate auf den 5k Evaluationsdaten deutlich gesteigert. Allerdings bringen morphem-basierte Erkenner auch eine höhere akustische Verwechselbarkeit der Erkennungseinheiten, sowie eine Reduktion der effektiven Reichweite statistischer n-gramm-Modelle mit sich, sodass die Qualität bei größeren Morphinventaren unter den Erwartungen blieb.

Wenn Morpheme als Basiseinheiten zur Sprachmodellierung verwendet werden, können Abhängigkeiten auf der Wortebene verloren gehen. Als Lösung für dieses Problem wurden Modelle auf der Basis der Wortwurzel (root) im Rahmen von faktorisierten Sprachmodellen untersucht. Zwar ergeben sich auf dieser Modellierungsgrundlage erheblich bessere Werte für die Testdatenwahrscheinlichkeit, wegen der schwächeren Vorhersagekraft konnte eine Verbesserung der Worterkennungsrate aber nicht erreicht werden.

Zusätzlich zu den rein morphem-basierten Sprachmodellen wurden verschiedene faktorisierte Sprachmodelle entwickelt, die es gestatten, wort-basierte Modelle durch unterschiedliche morphologische Information anzureichern. Die Ergebnisse zeigen, dass die Verwendung solcher zusätzlicher Prädiktoren zu qualitativ besseren Modellen führt.

Mit der Arbeit konnte gezeigt werden, dass die Verwendung von Morphemen zur Modellierung morphologisch reichhaltiger Sprachen vorteilhaft ist und insbesondere zu einer Reduktion der OOV-Rate führt. In der Folge ergeben sich auch verbesserte Werte für die Worterkennungsrate von Spracherkennungssystemen mit kleinem Morpheminventar. Es hat sich herausgestellt, dass die Verwendung morphem-basierter Sprachmodelle als Zusatzkomponente in wort-basierten Modellen nutzbringend ist. Die Arbeit hat auch bestätigt, dass die Evaluation von Sprachmodellen stets durch Einbettung in diejenigen Anwendung erfolgen sollte für die es entwickelt wurde. Nur so kann derzeit zuverlässig ermittelt werden, ob das Modell tatsächlich eine Verbesserung erbringt.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Natural Language Processing, also called Human Language Technology or Language Technology or Speech and Language Processing, is an interdisciplinary field which aims at getting computers perform useful tasks involving natural language such as enabling human-machine communication, improving human-human communication, or simply doing useful processing of text and speech [Jurafsky and Martin, 2008]. Research and development activities in this field include the coding, recognition, interpretation, translation, and generation of human languages [Cole et al., 1997]. The end results of such activities are speech and language technologies such as speech recognition and synthesis, machine translation, text categorization, text summarization, information/text retrieval, information extraction, etc.

Many speech and language technologies can be formulated as a problem in communication theory, particularly as a source-channel or noisy-channel model. A source-channel model has first been exploited for continuous speech recognition by the IBM speech recognition group [Bahl et al., 1983]. However, later it has also been applied in many other natural language processing applications including machine translation, spelling correction, optical character recognition, etc. This model basically states that a communication channel is a system in which the output depends statistically on the input. It is characterized by a conditional probability distribution $p(Y|W)$ that Y emerges from the channel given W was input. In automatic speech recognition, Y is an acoustic signal; in machine translation, Y is a sequence of words in the target language; and in spelling correction, Y is a sequence of characters produced by a possibly imperfect typist [Brown et al., 1992]; in optical character recognition, Y is the image of the printed characters; and in handwriting recognition, Y is the sequence of strokes on a tablet [Roukos, 1997]. In all of these applications, we face the problem of recovering a string of words after it has been "distorted" by passing through a noisy-channel. Part of tackling this problem is to estimate the probability with which any particular string of words will be presented as input to the noisy-channel [Brown et al., 1992]. This is the task of language modeling. Therefore, a language model is a key component in speech and language processing and research on language modeling serves a wide spectrum of applications [Roukos, 1997]. The following section discusses language modeling.

1.1 Language Modeling

The goal of language modeling is to characterize, capture and exploit the restrictions imposed on the way in which words can be combined to form sentences and by doing so to describe how words are arranged in a natural language. It is an attempt to capture the inherent regularities (in word sequence) of a natural language. Language modeling is fundamental to many natural language applications. Having good language models is, therefore, important to improve the performance of speech and natural language processing systems.

Historically, two major techniques have been used to model languages. The first one relies on grammars, such as context free grammars or unification grammars, which are defined based on linguistic knowledge. The second technique uses corpus based probabilistic models which are widely applied in natural language processing since their introduction in the 1980's. The scope of this thesis is limited to the latter models, namely, statistical language models (SLM).

Although SLMs have first been introduced for speech recognition systems [Jelinek, 1990], in principle, they can be used in any natural language processing application where the prior probability of a word sequence is important. Thus, they are used in statistical machine translation [Brown et al., 1990, 1993], spelling correction [Keringhan et al., 1990, Church and Gale, 1991], character and handwriting recognition [Kopec and Chou, 1994, Kolak and Resnik, 2002], part-of-speech tagging [Bahl and Mercer, 1976, Church, 1988], text summarization and paraphrasing [Knight and Marcu, 2002, Barzilay and Lee, 2004], question answering [Echihabi and Marcu, 2003], and information retrieval [Croft and Lafferty, 2003].

Statistical language models provide an estimate of the probability of a word sequence W for a given task. If W is a specified sequence of words, $W = w_1, w_2, \cdots, w_q$ then it would seem reasonable that $P(W)$ can be calculated as:

$$p(W) = p(w_1)p(w_2|w_1)p(w_3|w_1w_2), \cdots, p(w_q|w_1w_2, \cdots, w_{q-1}) \tag{1.1}$$

However, it is essentially impossible to estimate the conditional probabilities for all words and all sequence lengths in a given language. Thus, n-gram models, in which the conditional probabilities are approximated based only on the preceding $n-1$ words, are used. Even n-gram probabilities are difficult to estimate reliably for all n.

The probabilities in n-gram models are commonly determined by means of maximum likelihood estimation (MLE). This makes the probability distribution dependent on the available training data and on how the context has been defined [Junqua

and Haton, 1996]. Thus, to ensure statistical significance, large training data are required in statistical language modeling [Young et al., 2006]. Even if we have a large training corpus, there might still be many possible word sequences which will not be encountered at all, or which appear with a statistically insignificant frequency (data sparseness problem) [Young et al., 2006]. Moreover, there might also be individual words which are not in the corpus at all - Out-of-Vocabulary (OOV) words problem. Words or word sequences not observed in a training data will be assigned zero probability while those words or word sequences that appeared with a statistically insignificant frequency will be assigned a poor probability estimate. This negatively affects the performance of an application that uses the language model. Smoothing techniques (discussed in detail in chapter 2) and several other modeling techniques such as class based language models are used to alleviate the problem of OOV and data sparsity. Nevertheless, the problems of data sparseness and out-of-vocabulary words are still challenging for morphologically rich languages like Amharic.

1.2 Natural Language Processing for Amharic

Research and development in the area of human language technology have been and are being conducted for more than 50 years with a particular focus on few languages, and consequently most of the technologies are available for technologically favoured languages like English, French, German or Japanese. Although language technologies are equally important and useful for under served languages such as the Ethiopian ones, research in this field started only recently and no usable basic technologies are available for these languages. On the other hand, to survive in the information age and help people in their development, Ethiopian languages require to have basic technological tools [Atelach et al., 2003]. Thus, there is an urgent need for a variety of natural language applications including spell-checkers, machine translation systems, natural language generator, speech recognizer and synthesizer, information retrieval system etc. for Ethiopian languages in general and Amharic in particular [Atelach et al., 2003].

To this end, various Masters and Ph.D. students conducted research in natural language processing specifically in the area of Treebank, electronic thesaurus and dictionary construction, stemming, morphological analysis and synthesis, part of speech tagging, natural language (sentence) parsing, data driven bilingual lexical acquisition, machine translation, indexing, information retrieval, data mining, text classification and categorization, summarization, spelling checker, optical character recognition, speaker verification, speech recognition and synthesis. Although language models are fundamental for many of the natural language processing ap-

plications, so far they did not receive the attention of researchers who are working on Ethiopian languages. Even those who worked in the area of speech recognition (Solomon [2001], Kinfe [2002], Zegaye [2003], Martha [2003], Molalgne [2004], Hussien and Gambäck [2005] and Solomon [2006]), to which a language model is central [Jelinek, 1990], concentrated on the acoustic modeling part and paid little or no attention to language modeling. These works, therefore, could not achieve the performance improvement which can be obtained using a high quality language model. Furthermore, since Amharic is a morphologically rich language, language modeling for Amharic is demanding and deserves special attention.

1.3 Morphology of Amharic

Amharic is one of the morphologically rich languages. Like other Semitic languages, Amharic exhibits a root-pattern morphological phenomenon. A root is a set of consonants (also called radicals) which has a basic lexical meaning. A pattern consists of a set of vowels which are inserted among the consonants of a root to form a stem. In addition to this non-concatenative morphological feature, Amharic uses different affixes to create inflectional and derivational word forms.

Some adverbs can be derived from adjectives. Nouns are derived from other basic nouns, adjectives, stems, roots, and the infinitive form of a verb by affixation and intercalation. Case, number, definiteness, and gender marker affixes inflect nouns.

Adjectives are derived from nouns, stems or verbal roots by adding a prefix or a suffix. Moreover, adjectives can also be formed through compounding. Like nouns, adjectives are inflected for gender, number, and case [Baye, 2000EC].

Amharic verbs are derived from roots. The conversion of a root to a basic verb stem requires both intercalation and affixation. For instance, from the root gdl 'kill' we obtain the perfective verb stem gäddäl- by intercalating pattern ä_ä. From this perfective stem, it is possible to derive a passive (tägäddäl-) and a causative stem (asgäddäl-) using prefixes tä- and as-, respectively. Other verb forms are also derived from roots in a similar fashion. Verbs are inflected for person, gender, number, aspect, tense and mood [Baye, 2000EC]. Other elements like negative markers also inflect verbs in Amharic.

A more detailed description of Amharic morphology is given in Chapter 3. But from the above brief description, it can already be seen that Amharic is a morphologically rich language. It is this feature that makes development of language models for Amharic challenging.

1.4 Statement of the Problem

The data sparseness problem in statistical language modeling is more serious for languages with a rich morphology such as Amharic, Arabic, Hebrew or Turkish than for languages like English. Languages with rich morphology have a high vocabulary growth rate which results in high perplexity and a large number of out-of-vocabulary words [Vergyri et al., 2004]. The simplest or easiest solution for these problems might be extending the size of the training vocabulary so as to minimize OOV words. However, this is not efficient even for morphologically poor languages, let alone for morphologically rich languages as a vocabulary of any size becomes inadequate. Sproat [1992] stated that "... one can have a very large list of words yet still encounter many words that are not to be found in that list". He emphasized the problem of out-of-vocabulary words even for English, which is known to be a morphologically impoverished language. One can imagine how severe the data sparseness and OOV words problems are in morphologically rich languages. Thus, a way for building high quality language models on the basis of insufficient training material has to be found [Geutner, 1995]. A promising direction is to abandon the word as a modeling unit and split words into smaller word fragments [Hirsimäki et al., 2005].

Since Amharic is a morphologically rich language, an Amharic language model suffers from the data sparseness and out-of-vocabulary words problems. The negative effect of the Amharic morphology on language modeling has already been reported by Solomon [2006], who also recommended the development of sub-word based language models for Amharic.

The purpose of this study is, therefore, to explore the best way of modeling the Amharic language using morphemes as a language modeling unit. To this end, the research answers the following questions.

1. Can language modeling (for Amharic) be done on the sub-word level?
2. Which smoothing technique leads to the best quality language model?
3. Which kind of morphemes (statistical or linguistic ones) is better for modeling Amharic?
4. Is explicit treatment of the root-pattern morphology superior to simply ignoring it?
5. How to capture word level dependencies in morpheme-based language modeling?

6. Are morpheme-based language models superior to the word-based ones?
7. How to compare word-based and morpheme-based language models?
8. How can morpheme-based language models be used in a speech recognition system?
9. Do morpheme-based language models yield a performance improvement in a speech recognition system?
10. Does the integration of morphological features to word-based language models result in better quality models?

1.5 Contribution of the Research

Considering the main purpose of this research work, the following can be considered the major contributions of the study.

We showed the possibility of developing language models for Amharic using subwords or morphs as a solution to the data sparseness problem. Different morpheme-based language models have been developed. The results of our experiments confirmed that linguistic morphemes are better suited to model the Amharic language than the statistical ones and that an explicit treatment of root-pattern morphology is advantageous to simply ignoring it.

We further demonstrate that language models that have a different number of tokens and a different out-of-vocabulary rate can be best compared by integrating them into an application for which they have been designed.

We showed two ways of using morpheme-based language models in speech recognition. The first is a lattice rescoring approach where the language models were used to rescore word-based lattices. The second approach uses morphemes as lexical entries and language model units. Here, the speech recognition system recognizes morphemes and sequences of morphemes need to be concatenated into words. Morpheme-based language models yielded a performance improvement using both methods except for the latter if applied to large vocabularies.

Since Amharic roots represent the lexical meaning of words, we developed root-based language models which are able to capture word level dependencies in a morpheme-based language model.

Several factored language models that integrate an extra word feature to language models have also been developed and found to be high quality models in terms of perplexity and word recognition accuracy .

It has also been found that among a range of smoothing techniques, Kneser-Ney and its variations produced better language models regardless of the language modeling unit used.

1.6 Scope and Limitation of the Study

The scope of this research is limited to statistical language modeling. The development of statistical language models requires a text corpus consisting of millions of sentences. In this regard, it is obvious that the amount of data used in our experiment is small. Although it might be possible to take text from the web, the normalization task is laborious as, to our knowledge, there are no text processing tools for the language.

The unavailability of a rule based Amharic morphological analyzer that serves our purpose was a major limitation since the study should explore the effect of explicit treatment of root-pattern morphology on the quality of morph-based language models. Due to the arduous task of manual segmentation, we prepared a linguistically segmented corpus consisting of only 21,338 sentences. Although the amount of data used is very small, it was possible to explore the advantage of explicit treatment of the Amharic root-pattern morphology.

1.7 Organization of the Thesis

The thesis is organized into seven chapters. This chapter gives background information and states the problems which are addressed in the study. Chapter two presents the fundamentals of language modeling and a review of prominent works in morpheme-based language modeling. The nature of Amharic morphology are presented in chapter three. Chapter four provides a brief overview of computational morphology and a review of works on Amharic computational morphology. Chapter five presents details of the morphology-based language modeling experiments and the results of the experiments are also discussed in this chapter. Chapter six deals with the speech recognition experiments (lattice rescoring and morpheme-based speech recognition) and presents the results. Chapter seven contains conclusions and recommendations for future work.

Chapter 2

Fundamentals of Language Modeling

In Chapter 1, we indicated that SLMs can be applied in any natural language application where the prior knowledge of admissible word sequences in a natural language is important. Here we explain how language models are used in a natural language application, taking the speech recognition problem as an example since we consider automatic speech recognition as one of the most important application areas. However, as the focus of the thesis is on language modeling and not on speech recognition in general, we do not delve into this topic. The chapter also provides a description of evaluation metrics for language models, training procedures for computing the probabilities, smoothing techniques and the various kinds of language models.

2.1 Language Models in Speech Recognition

A speech recognition system is a system that automatically transcribes speech into text. The task can be formulated as a noisy-channel model and as a result, the speech recognition problem is considered a special case of Bayesian inference. Thus, the problem of speech recognition can be seen as that of finding the sequence of words $\hat{\mathrm{W}}$ that maximizes the conditional probability $p(W|A)$, which denotes the probability that the words W are spoken given the observed acoustic evidence A, i.e.

$$\hat{W} = \arg\max_{w} p(W|A) \tag{2.1}$$

This probability is computed using Bayes theorem as:

$$\hat{W} = \arg\max_{w} \frac{p(A|W)p(W)}{p(A)} \tag{2.2}$$

where $p(A)$ is the average probability that A (the acoustic evidence) will be observed, $p(A|W)$ is the probability that the acoustic evidence A will be observed given the

speaker says W, and $p(W)$ is the prior probability that the word string W will be uttered [Jelinek, 1997].

Since the $p(A)$ is fixed for each sentence (sequence of word), it can be ignored. Consequently, the recognizer's task can be reduced to that of finding $\hat{W} = \arg\max_w p(A|W)p(W)$. This makes the probabilities $p(A|W)$ and $p(W)$ to be central in a speech recognition system. Figure 2.1 shows how these probabilities are integrated to recognize a sentence.

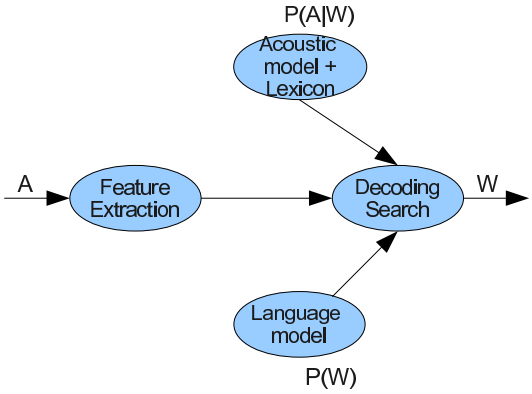

Figure 2.1: Components of a speech recognizer (taken from Jurafsky and Martin [2008])

Figure 2.1 also indicates the processes/components of a speech recognition system. The speech extraction includes sampling the electric signal, which is the output of a microphone, and transforming it into spectral features. The acoustic model, which also includes the word pronunciation dictionary, computes the probability $p(A|W)$ while the language model computes the prior probability of the word sequence W, $p(W)$. Finally, the decoder searches for a sequence of words that maximizes the product of the probabilities given by the acoustic and the language models. Alternatively, the decoder may provide N hypotheses, represented either as an N-best list or a lattice, instead of a single most likely hypothesis. We do not explain all the components in detail but concentrate only on language modeling.

2.2 Language Modeling

A statistical language model (SLM) is a probability distribution p(W) over word sequences W that models how often each sequence W occurs as a sentence. For a word sequence W ($W = w_1, w_2, \cdots, w_q$), the probability $P(W)$ can be calculated using the chain-rule as:

$$
\begin{aligned}
p(W) &= p(w_1 w_2, \cdots, w_q) \\
&= p(w_1)p(w_2|w_1)p(w_3|w_1 w_2), \cdots, p(w_q|w_1 w_2, \cdots, w_{q-1}) \\
&= \prod_{i=1}^{q} p(w_i|w_1, \cdots, w_{i-1})
\end{aligned} \tag{2.3}
$$

However, it is essentially impossible to estimate the conditional probabilities, $p(w_q|w_1 w_2, \cdots, w_{q-1})$ for all words and all sequence lengths in a given language since most histories would be unique or would have appeared only a few times. In practice, the estimation of these probabilities is infeasible because the set of strings is infinite. Thus, the Markov assumption, that says we can predict the probability of some future unit by only looking into the most recent observations, is applied. The most widely used language models, namely n-grams, are based on this assumption. Instead of computing the probability of a word given its entire or long history, we will approximate the history by just the last few words [Jurafsky and Martin, 2008]. That means $p(w_q|w_1 w_2, \cdots, w_{q-1})$ is approximated as:

$$
p(w_q|w_1 w_2, \cdots, w_{q-1}) = p(w_q|w_{q-n+1}, \cdots, w_{q-1}) \tag{2.4}
$$

based only on the preceding $n-1$ words. However, as n increases the number of n-grams also increases and the models become complex in terms of the number of parameters they employ. Because of their memory requirement and sparseness of training data, n is usually limited to 2, 3 or possibly 4.

2.2.1 Probability Estimation

The probabilities in n-gram models are commonly estimated based on **maximum likelihood estimation (MLE)** - that is by counting events in context on some training corpus. That means we take counts from a corpus and normalize them so that they lie between 0 and 1 [Jurafsky and Martin, 2008] as shown in the following equation.

$$
p(w_q|w_{q-N+1}, \cdots, w_{q-1}) = \frac{C(w_q|w_{q-N+1}, \cdots, w_q)}{C(w_q|w_{q-N+1}, \cdots, w_{q-1})} \tag{2.5}
$$

where C(.) is the amount of a given word sequence in the training data. From equation 2.5, we can see that the n-gram probability is computed by dividing the observed count of an n-gram with the observed count of its prefix.

Maximum likelihood estimation is solely dependent on the training data. That

means we can get a good probability estimate for those n-grams that occurred in the training data in a sufficient number of times. However, there might be n-gram sequences that may not exist in our corpus. These n-grams will, therefore, be assigned zero probabilities even if they are legal sequences in a given language. Consequently, speech and language processing applications using such a model will decide erroneously. In addition, even if a certain sequence has a non zero-count, the probability estimate might be poor if the count is small. Therefore, a method that can help to obtain better probability estimates is required. Smoothing is usually used for this purpose. Section 2.4 gives descriptions of the various smoothing techniques developed so far. Before that we would like to discuss how language models are evaluated.

2.3 Evaluation of Language Models

The best way of evaluating language models is measuring its effect on a specific application for which it was designed [Rosenfeld, 1997]. This way of evaluation is called extrinsic evaluation. However, it is computationally expensive, hard to measure and bad for a task independent comparison. There is, therefore, a need for an intrinsic evaluation metric which measures the quality of language models independent of any application. The commonly used way is to evaluate a language model by the probability it assigns to some unseen text (test set), a text which is not used during model training. Better models will assign a higher probability to the test data [Jurafsky and Martin, 2008]. Based on this probability, two related intrinsic evaluation metrics - Cross entropy and Perplexity - are computed.

2.3.1 Cross-entropy

The cross-entropy of a model m on some distribution p is defined as:

$$H(p,m) = \lim_{n\to\infty} -\frac{1}{n}\sum_{w\in L} p(w_1,...,w_n)\log m(w_1,...,w_n) \tag{2.6}$$

For a stationary and ergodic process, we can estimate the cross-entropy by taking a single sequence that is long enough instead of summing over all possible sequence as follows.

$$H(p,m) = \lim_{n\to\infty} -\frac{1}{n}\log m(w_1,...,w_n) \tag{2.7}$$

As we can notice from the above equation, cross-entropy is defined in the limit. But, with a sufficiently large sequence of fixed length, we can approximate cross-entropy

of a model $M = P(w_i|w_{i-N+1...w_{i-1}})$ on a sequence of words W as [Jurafsky and Martin, 2008]:

$$H(W) = -\frac{1}{N}\log P(w_1 w_2 ... w_N) \tag{2.8}$$

Where, N is the number of tokens in a test text.

This measures the average surprise of a model in seeing the test set and the aim is to minimize this number. Cross entropy is inversely related to the probability assigned (by the model) to the word sequence of the test data. That means a high probability leads to a low cross entropy.

2.3.2 Perplexity

A related most commonly used intrinsic language model evaluation metric is perplexity. Perplexity is computed as:

$$\begin{aligned} PP(W) &= 2^{H(W)} \\ &= P(w_1 w_2 ... w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 ... w_N)}} \\ &= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 ... w_{i-1})}} \end{aligned} \tag{2.9}$$

Perplexity can be interpreted as the average branching factor of a language model. Models with low perplexity values for a given data set are better models. As can be seen from equation 2.9, a higher conditional probability of the word sequence leads to lower perplexity. Thus, minimizing perplexity is equivalent to maximizing the test set probability [Jurafsky and Martin, 2008].

Since the calculation of both cross entropy and perplexity is based on the number of tokens in a test set, vocabularies must be the same when perplexities or cross entropies are compared. Otherwise, the measures are not comparable. Even if the vocabularies are equal, different texts, with different out-of-vocabulary words rate will make the comparison dubious [Rosenfeld, 2000]. Some researchers [Gauvain et al., 1995, Hacioglu et al., 2003] used normalized perplexity for comparison purposes. However, they did not give an explanation about the soundness of the normalization method (formula) they used. When we have different token counts, models can perhaps be compared on the basis of the probability they assign to the test sets. In addition, improvement in perplexity does not always guarantee an im-

provement in the performance of the application for which the language model is used [Jurafsky and Martin, 2008]. Thus, several attempts have been made to devise metrics that are better correlated with the error rate of the application, especially the speech recognizer. Such metrics include speech decoder entropy [Ferretti et al., 1989], acoustic perplexity and synthetic acoustic word error rate [Axelrod et al., 2007]. Nevertheless, perplexity continues to be the preferred metric for practical language model construction [Rosenfeld, 2000].

2.4 Smoothing Techniques

Smoothing is a term that describes techniques for adjusting the MLE probabilities and to produce more accurate probabilities [Chen and Goodman, 1998]. The name - smoothing - comes from "the fact that (looking ahead a bit) we will be shaving a little bit of probability mass from the higher counts, and piling it instead of the zero counts, making the distribution a little less jagged" [Jurafsky and Martin, 2008]. Smoothing not only prevents zero probabilities, but also attempts to improve the accuracy of the model as a whole [Chen and Goodman, 1998]. The following are some of the smoothing techniques used in language modeling.

2.4.1 Laplace Smoothing

In Laplace smoothing, which is also called Laplace's Law or Add-one smoothing, we pretend that each n-gram occurs once more than it actually does. That means we add one to all the counts before we normalize them into probabilities. According to Laplace smoothing, the uni-gram probability of the word w_i is computed as follows:

$$p(w_i)_{addone} = \frac{c(w_i) + 1}{N + V} \tag{2.10}$$

where $c(w_i)$ is count of w_i, N is the total number of word tokens and V is the number of types, as opposed to the unsmoothed MLE of the uni-gram probability of the word w_i:

$$p(w_i) = \frac{c(w_i)}{N} \tag{2.11}$$

However, the estimates of Laplace law are dependent on the size of the vocabulary [Manning and Schütze, 1999]. If the data is sparse for a large vocabulary, Laplace law gives too much of the probability mass to unseen events[1]. That means, it does

[1]Laplace law gives very low probability to each unseen event. However, if the data is sparse over large vocabulary, there will be a lot of unseen events which makes the total probability mass allocated to them fairly large.

not perform well enough to be used in n-gram models [Jurafsky and Martin, 2008].

2.4.2 Add λ Smoothing

This smoothing method is introduced as a solution for the overestimation problem of Laplace smoothing. It is based on the assumption that a smaller probability mass will be moved to unseen events by adding a fractional count λ (which is between 0 and 1) rather than 1. Therefore, add λ smoothing (which is also called Lidstone's law) adds a positive value λ which is normally smaller than 1 to each n-gram count. That means the probability will be calculated as follows.

$$p(w_i)_{add\lambda} = \frac{c(w_i) + \lambda}{N + V\lambda} \tag{2.12}$$

Although this method avoids the limitation of Laplace smoothing by choosing a small λ, it has also its own drawbacks. It requires a method for choosing an appropriate value for λ dynamically [Jurafsky and Martin, 2008, Manning and Schütze, 1999]. Moreover, it always gives probability estimates linear in the MLE frequency and this is not a good match to the empirical distribution at low frequencies [Manning and Schütze, 1999]. Therefore, there was a need for better smoothing methods.

2.4.3 Natural Discounting

Natural discounting [Ristad, 1995] is introduced as a solution to the problems of Laplace and Add λ smoothing techniques. It is principally inspired by the theory of stochastic complexity and the theory of algorithmic complexity. Instead of estimating parameter values, natural discounting method imposes constraints on strings so that simple strings are more probable than complex ones. This smoothing technique considers the number of observed types and the vocabulary size in the probability calculation as shown below.

$$P_{ND}(w_i|w_{i-1}) = \begin{cases} \frac{C(w_{i-1}w_i)}{C(w_{i-1})}, & if\ q = k \\ \left(\frac{C(w_{i-1}w_i)}{C(w_{i-1})}\right) \frac{C(w_{i-1})(C(w_{i-1})+1)+q(1-q)}{C(w_{i-1})^2+C(w_{i-1})+2q}, & if\ q < k\ \&\ C(w_{i-1}) > 0 \\ \left(\frac{1}{k-q}\right) \frac{q(q+1)}{C(w_{i-1})^2+C(w_{i-1})+2q}, & otherwise \end{cases} \tag{2.13}$$

where q is the number of observed types, k is the vocabulary.

2.4.4 Good-Turing Smoothing

The Good-Turing smoothing was first described by Good [1953]. The idea of this algorithm is to re-estimate the amount of probability mass to assign to n-grams that occurred zero times by looking at the number of n-grams that occurred once, i.e. based on the number of singletons or hapax legomena [Jurafsky and Martin, 2008]. This smoothing technique is based on computing N_c frequency of frequency or count of count. For example, assuming bi-grams, N_0 is the number of bi-grams with count 0, N_1 the number of bi-grams with count 1, and so on. The Good-Turing estimate replaces an MLE count c with a corrected count c^*, which is calculated as a function of N_{c+1}:

$$c^* = (c+1)\,\frac{N_{c+1}}{N_c} \tag{2.14}$$

This way we can replace all the MLE counts and the probability can be calculated as $p = \frac{c^*}{N}$, where N is the total number of counts in the distribution. Alternatively, instead of computing the corrected count c^* for N_0, we calculate the probability of events that had zero count N_0 as follows:

$$P^*_{GT} = \frac{N_1}{N} \tag{2.15}$$

where, N_1 is the number of items that occurred once and N is the total number of items we have seen in the training data. This probability mass is then distributed among unseen events uniformly, or by some other sophisticated method [Manning and Schütze, 1999].

Good-Turing method assumes that we know N_0 (the number of missing n-grams) and the distribution of words and n-grams is binomial, although this is not actually the case. In addition, Good-Turing estimate can not be used when the count of count is 0 [Jurafsky and Martin, 2008]. Therefore, there is also a need to smooth the count of counts so that they are all above zero. Gale and Sampson [1995] proposed a simple and effective algorithm as a solution, called Simple Good-Turing. In this method, the count of counts N_c are smoothed before they are used in the computation of the corrected counts.

Since large counts are assumed to be reliable, the discounted/corrected count c^* is not used for all counts. Moreover, it is customary to treat n-grams with low counts as if the count were 0. Thus, in practice Good-Turing discounting is not used by itself, but in combination with backoff and interpolation algorithms [Jurafsky and Martin, 2008].

2.4.5 Interpolation and Backoff

The smoothing techniques discussed so far solve the problem of data sparsity based on the raw frequency of n-grams. Another way of approaching the problem is based on the n-gram hierarchy. That is, we estimate a probability $p(w_n|w_{n-1}w_{n-2})$ of a tri-gram, for which no example is found in our training data, based on bi-gram probability $p(w_n|w_{n-1})$, and so on. There are two ways of using the n-gram hierarchy: interpolation and backoff. The following is a brief description of these methods.

2.4.5.1 Interpolation

In interpolation, we combine the probability estimates of all n-gram orders based on the assumption that if there is insufficient data to estimate a probability in the higher-order n-gram, the lower order can often provide useful information. In simple linear interpolation (which is also called **Jelinek-Mercer smoothing** after its developers Jelinek and Mercer [1980]), we estimate the tri-gram probability $p(w_n|w_{n-1}w_{n-2})$, for example, by mixing together the uni-gram, bi-gram, and tri-gram probabilities, each weighted by a λ as follows:

$$P^*(w_n|w_{n-1}w_{n-2}) = \lambda_1 p(w_n|w_{n-1}w_{n-2}) + \lambda_2 p(w_n|w_{n-1}) + \lambda_3 p(w_n) \qquad (2.16)$$

where

$$\sum_i \lambda_i = 1 \qquad (2.17)$$

In a slightly more sophisticated version of linear interpolation, each λ weight is computed based on the context/history. For example, if the context of a certain tri-gram is observed frequently, then a high λ will be suitable for the tri-gram, and consequently we give more weight to the tri-gram in the interpolation. On the contrary, for a history that has occurred only once, a lower λ will be appropriate [Chen and Goodman, 1998].

In both the simple and conditional interpolation, the λ_s are learned from some data using algorithms such as Baum-Welch, or Expectation-Maximization. We choose the λ values that maximize the likelihood of some data that is different from the training corpus. One can use for instance a held-out corpus or the technique known as deleted interpolation. In deleted interpolation different parts of the training data rotate in training either the maximum likelihood probabilities or the λ_s and the average of the results is then used [Chen and Goodman, 1998].

2.4.5.2 Backoff

Another way of using information from the n-gram hierarchy to obtain better estimates is backoff. In this method, probabilities are computed based on higher order n-gram counts if the counts are nonzero. Otherwise, we compute the probability of a particular n-gram based on (n-1)-gram. If the count of the (n-1)-gram is also zero, then we backoff to the (n-2)-gram. We continue backing off until we reach a history that has some counts. Katz backoff is one example of such a model.

Katz Smoothing [Katz, 1987] is a backoff smoothing with Good-Turing discounting. The method extends the intuition of Good-Turing estimate by adding information from n-gram hierarchy. It uses discounting to know how much total probability mass to set aside for all unseen events, and backoff to distribute this probability. This method provides a better way of distributing the probability mass among unseen n-grams on the basis of the information it gets from lower order n-grams. Katz backoff generally works as follows: if we have a nonzero count for an n-gram, then it relies on the discounted/corrected (according to Good-Turing smoothing) probability P^*. Otherwise, we recursively backoff to the Katz probability for the shorter history [Jurafsky and Martin, 2008]. The following formulae show how tri-gram and bi-gram probabilities are computed:

$$P_K\left(w_n|w_{n-1}w_{n-2}\right) = \begin{cases} P^*\left(w_n|w_{n-1}w_{n-2}\right), & if\ C\left(w_{n-2}w_{n-1}w_n\right) > 0 \\ \alpha\left(w_{n-2}w_{n-1}\right) P_K\left(w_n|w_{n-1}\right), & else\ if\ C\left(w_{n-2}w_{n-1}\right) > 0 \\ P^*\left(w_n\right), & otherwise. \end{cases} \tag{2.18}$$

$$P_K\left(w_n|w_{n-1}\right) = \begin{cases} P^*\left(w_n|w_{n-1}\right), & if\ C\left(w_{n-1}w_n\right) > 0 \\ \alpha\left(w_{n-1}\right) P^*\left(w_n\right), & otherwise. \end{cases} \tag{2.19}$$

where $P^*\left(.\right)$ is a discounted probability and α is a weight used to ensure that the probability mass for the lower order n-grams sums up to exactly the amount that we saved by discounting the higher-order n-grams. Details of the computation of $P^*\left(.\right)$ and α can be found in Jurafsky and Martin [2008].

2.4.6 Witten-Bell Smoothing

Witten-Bell smoothing [Witten and Bell, 1991], which can be considered as an instance of linear interpolation, was first developed for the task of text compression. In this method, the nth-order smoothed model is defined recursively as a linear in-

terpolation of nth-order MLE model and the (n-1)th-order smoothed model [Chen and Goodman, 1998] as it is given below for a tri-gram model.

$$\begin{aligned} P_{WB}\left(w_n|w_{n-2}w_{n-1}\right) = \\ \lambda_{w_{n-2}w_{n-1}} P_{MLE}\left(w_n|w_{n-2}w_{n-1}\right) \\ + \left(1-\lambda_{w_{n-2}w_{n-1}}\right) P_{WB}\left(w_n|w_{n-1}\right) \end{aligned} \tag{2.20}$$

where the probability mass given to unseen n-grams is calculated on the basis of the number of unique words that follow the history($w_{n-2}w_{n-1}$), which is the number of observed tri-gram types.

$$1-\lambda_{w_{n-2}w_{n-1}} = \frac{|\left\{w_i : C(w_{i-2}w_{i-1}w_i) > 0\right\}|}{|\left\{w_i : C(w_{i-2}w_{i-1}w_i) > 0\right\}| + N} \tag{2.21}$$

where N is the number of tokens that follow the history ($w_{n-2}w_{n-1}$). Chen and Goodman [1998] found out that this smoothing method performs poorly specially when used on small training sets.

2.4.7 Absolute Discounting

In absolute discounting [Ney and Essen, 1991, Ney et al., 1995] all non-zero MLE counts are discounted by a small constant amount D. The discounted frequencies are then distributed over unseen events [Manning and Schütze, 1999]. The idea was, if we have good estimates for higher counts, a small discount won't affect them [Jurafsky and Martin, 2008]. The probability is computed as follows.

$$P_{abs}\left(w_n|w_{n-1}\right) = \begin{cases} \frac{C(w_{n-1}w_n)-D}{(w_{n-1})}, & if\ C(w_{n-1}w_n) > 0 \\ \alpha(w_{n-1})P_{abs}(w_n), & if\ C(w_{n-1}w_n) = 0 \end{cases} \tag{2.22}$$

where $C(.)$ is the count of the n-gram, D is the discounting value which is between 0 and 1 ($0 \leq D \leq 1$) and α is the backoff weight, which is calculated as:

$$\alpha(w_{n-1}) = \frac{1-\sum_n P_{abs}(w_n|w_{n-1})}{1-\sum_n P_{abs}(w_n)} \tag{2.23}$$

Absolute discounting can also be used in the framework of interpolation as follows.

$$P_{abs}(w_n|w_{n-1}) = \frac{max\{C(w_{n-1}w_n)-D, 0\}}{C(w_{n-1})} + \beta(w_{n-1})P_{abs}(w_n) \tag{2.24}$$

where the interpolation coefficient β is calculated as:

$$\beta(w_{n-1}) = \frac{DN(w_{n-1}*)}{C(w_{n-1})} \tag{2.25}$$

where $N(w_{n-1}*)$ represents the number of unique words that follow the history w_{n-1}. In all cases, the value of D can be determined on some held out data or using cross validation technique on the training data. Ney et al. [1995] obtained the following optimal estimate for D through cross validation on the training data.

$$D = \frac{n_1}{n_1 + 2n_2} \tag{2.26}$$

where n_1 and n_2 are the total number of n-grams with exactly one and two counts, respectively, in the training data and n is the order of the higher-order model.

Absolute discounting is the base for the Kneser-Ney smoothing technique and its variant, which are known for their superior performance compared with other smoothing algorithms.

2.4.8 Kneser-Ney Smoothing

As indicated in section 2.4.7, Kneser-Ney smoothing is based on absolute discounting technique but uses a more sophisticated way to handle the backoff distribution. For example, consider the job of building a bi-gram model using a corpus which consists of common words such as Francisco, which occurs more often after a single word San. As the count of the word Francisco is high, the uni-gram probability is also high, and therefore, absolute discounting will give a relatively high probability to Francisco occurring after a new bi-gram history. However, this should not be the case as the word Francisco follows only the word San. Kneser-Ney smoothing tries to solve this problem. The main idea of this method is that the uni-gram probability should not be proportional to the frequency of a word, but to the number of different words that it follows [Chen and Goodman, 1998]. Thus, Kneser-Ney smoothing bases the estimate on the number of different contexts word w has appeared in assuming that words which have appeared in more contexts are more likely to appear in some other new context as well [Jurafsky and Martin, 2008]. Thus the uni-gram probability is defined as follows:

$$P(w_i) = \frac{|\{w_{i-1} : C(w_{i-1}w_i) > 0\}|}{\sum_{w_i} |\{w_{i-1} : C(w_{i-1}w_i) > 0\}|} \tag{2.27}$$

Assuming a proper coefficient α on the backoff so as to make everything sum to one, Kneser-Ney bi-gram backoff probability is given as follows:

$$P_{KN}\left(w_i|w_{i-1}\right) = \begin{cases} \frac{C(w_{i-1}w_i)-D}{C(w_{i-1})}, & if\ C(w_{i-1}w_i) > 0 \\ \alpha(w_i)\frac{|\{w_{i-1}:C(w_{i-1}w_i)>0\}|}{\sum_{w_i}|\{w_{i-1}:C(w_{i-1}w_i)>0\}|}, & Otherwise \end{cases} \tag{2.28}$$

The interpolated version of Kneser-Ney, which is called interpolated Kneser-Ney can be computed as shown below:

$$P_{KN}\left(w_i|w_{i-1}\right) = \frac{C(w_{i-1}w_i) - D}{C(w_{i-1})} + \beta(w_i)\frac{|\left\{w_{i-1} : C(w_{i-1}w_i) > 0\right\}|}{\sum_{w_i}|\left\{w_{i-1} : C(w_{i-1}w_i) > 0\right\}|} \tag{2.29}$$

This method is the most commonly used and the best performing modern n-gram smoothing method. It gave rise for the birth of its variant called Modified Kneser-Ney [Chen and Goodman, 1998].

2.4.9 Modified Kneser-Ney Smoothing

Modified Kneser-Ney smoothing was introduced by Chen and Goodman [1998]. This smoothing method differs from Kneser-Ney since it uses three different discounts D_1, D_2, D_{3+} that are applied to n-gram with one, two, and three or more counts, respectively instead of a single discount D for all nonzero counts. Chen and Goodman [1998] did the modification based on their observation that the ideal average discount for n-grams with one or two counts is substantially different from the ideal average discount for n-grams with 3 or more counts. They were able to show that this method significantly outperforms the regular Kneser-Ney smoothing. A bi-gram probability will be computed as follows.

$$P_{MKN}\left(w_i|w_{i-1}\right) = \frac{C(w_{i-1}w_i) - D(C(w_{i-1}w_i))}{C(w_{i-1})} + \beta(w_{i-1}w_i)P_{MKN}(w_i) \tag{2.30}$$

where

$$D(c) = \begin{cases} 0, & if\ c = 0 \\ D_1 & if\ c = 1 \\ D_2 & if\ c = 2 \\ D_{3+} & if\ c \geq 3 \end{cases} \tag{2.31}$$

As Ney et al. [1995] developed an optimal estimate for D (for both absolute discounting and Kneser-Ney smoothing) as a function of training data, Chen and Goodman [1998] did the same to estimate the optimal values for D_1, D_2 and D_{3+}.

$$
\begin{aligned}
Y &= \frac{n_1}{n_1 + 2n_2} \\
D_1 &= 1 - 2Y\frac{n_2}{n_1} \\
D_2 &= 2 - 3Y\frac{n_3}{n_2} \\
D_3 &= 3 - 4Y\frac{n_4}{n_3}
\end{aligned} \tag{2.32}
$$

where the n_i's have the same meaning as in absolute discounting and Kneser-Ney smoothing. If one of the n_i's (count of counts) is zero, then it is not possible to use this smoothing technique.

$\beta(w_{i-1}w_i)$ is computed as follows so as to make the distribution to sum to 1.

$$
\beta(w_{i-1}w_i) = \frac{\sum_{j=1}^{3+} D_j |\{w_{i-1} : C_j(w_{i-1}w_i) > 0\}|}{\sum_{w_i} |\{w_{i-1} : C(w_{i-1}w_i) > 0\}|} \tag{2.33}
$$

2.5 Improved Language Models

Smoothing techniques have been developed to improve the quality of n-gram language models. Besides smoothing, several other methods have also been tried to improve language models. The probability of a word in n-gram models is normally computed based on the previous $N-1$ words. Therefore, n-gram models can not capture long distance dependencies. In addition, since they consider only the very immediate history, they can not adapt to the style or topic of the document [Lau et al., 1993] and, therefore, are considered static models. In this section we give a brief description of language models which have been developed as remedies to these and other related problems of n-gram modeling. However, we do not claim to be exhaustive in covering all the techniques.

2.5.1 Class-based Models

Class based or cluster n-gram models [Brown et al., 1992] are particularly introduced to tackle the problem of data sparsity. In class-based language models, the probability of a word w_i is computed by multiplying the conditional probability of the word's class given the preceding classes and the word given its class. Classes based on part-of-speech tags, morphological features of words, or semantic information have been tried [Niesler, 1997]. Moreover, models based on automatically derived classes have been trained [Brown et al., 1992]. Class based models led to a reduction in perplexity when linearly interpolated with word based n-gram models.

2.5.2 Higher Order N-gram

Higher order n-gram (quadro- or penta-gram) models are an obvious solution to the problem of tri-gram models not being able to capture long distance dependencies. However, most of the quadro- or penta-gram sequences might not appear in the training data because of data sparseness. Therefore, we need either to backoff to or interpolate with lower-order n-grams (tri-gram, bi-gram, uni-gram). In addition, as n increases the complexity of the model also increases. Nevertheless, Goodman [2001] developed n-gram models of n 1 to 10, as well as 20 to study the relationship between n-gram order and perplexity. Using 280 million words as training data a small improvement in perplexity has been obtained in 6-gram and 7-gram models over the penta-gram one. However, only the difference in perplexity between quadro-gram and penta-gram was significant. He also indicated the fact that penta-gram models provide a good trade off between computational resources and performance.

2.5.3 Decision Tree Models

Decision tree-based language models, which have been first introduced by Bahl et al. [1989], were suggested as a solution for the problem of long distance dependencies in n-gram models. Tree based models can partition the space of histories by asking questions about the history h at each of the internal nodes. The probability distribution p(w|h) is then computed based on the training data at each leaf. In the first attempt [Bahl et al., 1989], only a slight improvement in perplexity was obtained over the normal n-gram model although it involved a very time consuming tree-building process (many months of training). It was possible to obtain an appreciably lower perplexity when the tree-based model is interpolated with the normal n-gram model. Thus, the authors indicated the fact that the best way to benefit from a tree-based language model is by using it together with the n-gram model instead of replacing it. Rosenfeld [2000] said that it is likely that trees that significantly outperform n-grams exist, however, finding them is difficult due to computational and data sparseness reasons.

2.5.4 Skipping Models

Skipping n-gram models [Huang et al., Rosenfeld, 1994, Martin et al., 1999], also called distance n-grams, are those models in which the probabilities are computed for sequence of n words with gaps between them. That means the words no longer form a consecutive sequence. Considering tri-gram models, for instance, probability of w_n can be calculated as $p(w_n|w_{n-3}, w_{n-1})$, $p(w_n|w_{n-3}, w_{n-2})$, etc. Martin et al. [1999] used the idea to skip repetitions (I, and, the, and a), fillers (uh and um) or

modifiers (very) in the history of n-grams based on the assumption that skipping these words does not change the meaning of a phrase. Besides capturing long distance dependencies, this model can be used as a solution for the data sparseness problem caused by higher order n-gram models.

2.5.5 Dynamic Language Models

A dynamic or adaptive language model, which can be either trigger- [Lau et al., 1993, Rosenfeld, 1996] or cache-based [Kuhn, 1988, Kupiec, 1989, Kuhn and Mori, 1990, 1992, Jelinek et al., 1991], is one that changes its estimates as a result of seeing some of the text. In trigger-based models certain words of the vocabulary will be identified as triggers and the presence of these trigger words in the history modifies (increases or decreases) the distribution of the predicted/triggered words. Cache-based language models are based on the assumption that a word used in the recent past is much more likely to be used again. Thus, a cache model, which can be implemented either as a class-based [Kuhn and Mori, 1990, 1992, Jelinek et al., 1991] or tri-gram model [Jelinek et al., 1991], estimates the probability of a word from its recent frequency of use instead of its overall frequency in the training text. Interpolated with a normal tri-gram model, the cache-based language model resulted in a reduction in perplexity and a reduced word error rate of speech recognition. However, cache models are not appropriate for domains where the previous words are not exactly known. In a speech recognition application, for example, if cache models make error on earlier words, they do not have a way to correct it unless there is some way for users to interfere [Jurafsky and Martin, 2008].

2.5.6 Mixture Models

Mixture models or sentence mixture models [Iyer et al., 1994, Iyer and Ostendorf, 1999] are a simple variation of the standard n-gram model. They are based on the observation that there are different sentence types in a corpus which could be grouped by style, topic, or any other criteria. N-gram models are, therefore, estimated for each group of sentences and linearly interpolated either at the n-gram level or at the sentence level. When they are combined at the sentence level the approach is called sentence mixture model.

2.6 Morphology-based Language Modeling

Despite the development of smoothing techniques and other modeling techniques such as class-based language models, the data sparseness and out-of-vocabulary

words problems remained as challenges for morphologically rich languages. This is the reason behind the use of sub-word modeling units for language modeling instead of the usual units, namely words.

2.6.1 Sub-word Based Language Modeling

Many researchers [Geutner, 1995, Carki et al., 2000, Byrne et al., 2000, Whittaker and Woodland, 2000, Whittaker et al., 2001, Siivola et al., 2003, Hirsimäki et al., 2005, Kirchhoff et al., 2002, Kiecza and Waibel, 1999, Kwon, 2000, Choueiter et al., 2006, El-Desoky et al., 2009, Heintz, 2010] developed sub-word based language models using different approaches. The approaches differ in the choice of the sub-word units (which can be syllables, chunks or morphemes) or the word decomposition method they use (statistical method, linguistically motivated, etc.). In almost all of the cases sub-word based language models have been applied to an automatic speech recognition application.

A reduction in perplexity and in the out-of-vocabulary rate has been reported in most of the studies. However, the perplexity reduction did not always lead to an improvement in the performance of automatic speech recognition systems. This is due to the high acoustic confusability of short units.

Different approaches have been tried to improve the system performance while using sub-word units. One of the methods is the lattice or n-best list rescoring approach in which the sub-word units are used only in the language modeling component. Improvement in system performance has been reported as a result of rescoring lattices (n-best lists) using sub-word models interpolated with normal word-based models. The problem with this approach is that it is not possible to avoid the out-of-vocabulary words problem since word based dictionaries are used to generate the n-best list or the lattices. Another approach was to concatenate acoustically confusable sub-word units into longer units which can be acoustically better distinguished. This method also may not completely avoid the out-of-vocabulary words problem.

Another challenge of using sub-words in language modeling is related to the scope of the n-grams. That means, for example, if a word is segmented into five sub-word units and if we develop a penta-gram language model, the n-gram spans only a single word. Therefore, capturing word level dependencies becomes problematic. Researchers tried to solve this problem by using longer size n-grams like six- or seven-grams. Skipping models of the type used in Byrne et al. [2000] can also be used to tackle this problem.

The following section reviews the most influential works in morpheme-based language modeling.

2.6.2 Review of Previous Works

Geutner [1995] developed a language model for German by decomposing full word forms into individual morphemes and using morphs as units so as to produce a better large-vocabulary speech recognition system. In the experiment, she used different methods of word decomposition: strictly morpheme based (linguistically based, for example, "weggehen" is decomposed as "weg-geh-en"), not strictly linguistically oriented, decomposition into root form (this is something like removing suffixes, for instance "weggeh" is considered instead of "weggehen") and combination of the last two (not strictly linguistic based and decomposition into root form). A reduction in vocabulary size has been observed in all cases. Although the perplexity improvement varies, all morpheme based models have a lower perplexity than the word based language model. The language model developed using morphemes obtained by strictly linguistic decomposition showed a higher reduction in perplexity than the others. However, this language model did not improve the performance of the speech recognition system. That is why the author tried a decomposition method which is not strictly linguistically oriented. This method led to both a perplexity reduction and a slight improvement in recognition accuracy. However, the author did not provide details of the decomposition method and therefore it is difficult to explain the performance improvement of the speech recognition system. No performance improvement has been obtained as a result of using a root based decomposition. Combining both methods led to a perplexity reduction but not to an improvement in the performance of the speech recognition system.

Carki et al. [2000] developed a speech recognizer for Turkish using an automatic decomposition of words into syllables and merging them into larger units by defining word-positioned syllable classes. These units are used to train a quadro-gram language model. As a result, the out-of-vocabulary rate was decreased by 50%. However, because of acoustic confusability, the performance of speech recognition system decreased.

Byrne et al. [2000] developed a morph-based language model for Czech and used it in a large-vocabulary continuous speech recognition system. Words were decomposed into stems and endings and a morpheme bi-gram model has been developed using each morpheme as a unit. With this language model, it was possible to obtain a reduction in perplexity as well as an improvement (over 37%) in word recognition accuracy over the baseline system. The result obviously contradicts the findings of other researchers. But as the authors noted, this might be due to the nature of the language, the decomposition method or the task. As their language model simply considers morphemes as units, the probability of the stem is predicted based on the

previous ending i.e. $p(s_i|e_{i-1})$. This leads to a loss of word level dependencies as the ending gives information about grammatical features and not about the previous word. Considering this fact, the model was modified so that the probability of s_i is calculated as $p(s_i|s_{i-1})$ and the probability of e_i is calculated as interpolation of $p(e_i|s_i)$ and $p(e_i|e_{i-1})$ using an interpolation weight ε, where $0 \leq \varepsilon \leq 1$. Surprisingly, a better morpheme based model has been obtained with $\varepsilon = 0.0$. Byrne et al. [2001] applied the morpheme-based language models to a morpheme-based speech recognition. Although the baseline morpheme-based recognition system performed worse than the word-based recognition, a result which is nearly as good as the word-based system has been obtained by optimizing scaling factors and rescoring lattices with a tri-gram morpheme-based language model.

Whittaker and Woodland [2000] conducted a research focusing on the selection of sub-word units for n-gram modeling of Russian and English. The sub-word units are called particles by the authors since they denote any possible part of a word whether it is a single character or a whole word. Whittaker and Woodland [2000] used three word decomposition methods: affix stripping (based on string matching) and two greedy, data driven algorithms called particle selection algorithm (PSA) and word decomposition algorithm (WDA). PSA is designed to determine particle units that best model the training data. It uses the word uni-gram and bi-gram statistics from the training data and a list of all possible candidate particles of different lengths. Each particle is inserted in all words and a change in training likelihood is computed. The particle that gave the greatest reduction in perplexity is permanently added to the final set of particles (S) which initially contains all single characters that occur in words of the vocabulary. WDA is designed to determine word decompositions that best model the training data. Given an initial decomposition for each vocabulary word, it optimizes the decomposition of each word in turn. The optimal decomposition of a word is the one that maximizes the likelihood of the training data. In their experiments, they used fixed length character decomposition and affix striping as initial decomposition. The particles (obtained using different decomposition methods) have been used only in the language modeling component of a speech recognition system using a lattice rescoring framework. They developed various 6-gram language models and applied the language models in a speech recognition system. Particle models gave similar perplexity and word error rate (WER)[2] results as the word based models. However, a reduction in perplexity and a small reduction in WER have been obtained using an interpolation of particle and word based models. Their results also show that the data driven decomposition

[2]Although the particles are considered as a unit in language modeling, word level perplexity and WER have been reported.

algorithms led to a greater improvement, which can be attributed to the nature of the algorithms.

Whittaker et al. [2001] also investigated the use of particles in language and acoustic modeling. Their vocabulary independent speech recognition system has three components: the particle-based speech recognizer, a converter which translates particle hypotheses into a graph of word candidates, and a decoder which selects the highest scoring word sequence. As it can be expected, an increase in WER has been observed compared to a word bi-gram speech recognizer of comparable complexity. However, it was possible to cover 18 OOV words which are not part of the decoder's 65k vocabulary.

In another study Whittaker et al. [2001] compared word-based, class-based and particle-based language models. An improvement in perplexity has been obtained as a result of interpolating particle models with word and class based models. The authors also indicated the fact that the improvement with particle models suggests their capability of tackling the data sparsity problem.

Siivola et al. [2003] developed an unlimited vocabulary speech recognizer for Finnish using syllables and morphemes as units both in language and acoustic models. While the syllable lexicon has been produced using a reasonably simple rule set, an unsupervised morphological learning algorithm, namely Morfessor [Creutz and Lagus, 2005] (for a detailed description see Section 4.3.2.1), has been applied to segment words into morphs. They developed a tri-gram language model for each of the lexicons (word, syllable and morph) and found that the word-based language model has a lower perplexity than the syllable and morph-based ones. However, they calculated the perplexity of the syllable and morph-based language models by normalizing the log probabilities with the number of word tokens instead of the respective syllable and morph tokens. This approach does not seem to be correct as the probabilities are collected for syllable and morph tokens. Speech recognition results have been reported in WER, token error rate (ToER) and letter error rate (LER). The results show that morph-based models outperform all the others despite the use of morphemes in both acoustic and language models. This result contradicts the findings of other similar experiments.

In another study, Hirsimäki et al. [2005] compared the performance of automatically learned morphs (statistical) with linguistic morphs for a Finnish speech recognition task. Their aim was to find out whether the error rate reduction obtained in the previous study [Siivola et al., 2003] is due to the reduction in out-of-vocabulary rate or whether other ways of splitting words would also give good results. Therefore, they considered three lexical units: statistical morphs (obtained using Morfessor), words extended with phonemes as sub-word units and grammat-

ical morphs (obtained using linguistic rules). N-gram language models of order 2 to 7, smoothed with the Kneser-Ney smoothing technique, have been trained for each of the lexical units using the SRILM toolkit. The language models have been evaluated using cross-entropy and the results show that morpheme-based models are better than word-based models. Similar to the results of their previous study, the morph-based language models performed better in speech recognition than the word based models. Therefore, they concluded that both the grammatical and statistical morphemes seem to be a good choice for representing a very large vocabulary efficiently with a reasonable number of lexical units.

Kirchhoff et al. [2002] developed particle-based (similar to Whittakers's model [Whittaker and Woodland, 2000]) language model for Arabic, motivated by the work of Billa et al. [1997] who have got promising results by just separating the definite article from the following noun. Instead of decomposing words fully into component morphemes, they detached the possessive and object pronoun suffixes, definite article, negation and future markers and prepositional morphemes. The particle models have higher perplexity compared to the word-based model. However, they calculated the perplexity in the same way as Siivola et al. [2003] did, that means the log probabilities accumulated over the particles but normalized with the number of words instead of the number of particles. Small reduction in WER has been achieved as a result of rescoring n-best lists using a model which is an interpolation of particle and word models.

Creutz et al. [2007] analysed sub-word based language models in large vocabulary continuous speech recognition of four morphologically rich languages: Finnish, Estonian, Turkish and Egyptian Colloquial Arabic. Their aim was to compare morpheme and word based n-gram language models in automatic speech recognition across languages. The result of their experiment shows that the morph-based models perform better than the word based models except for Arabic, where the word model outperforms the morph based model. The best performance is observed for Finish data sets, which is explained by the speaker dependent acoustic models and clean noise conditions. The authors attributed the poor performance of the Arabic setup to the insufficient amount of language model training data. However, we also note that the type of speech data used to train the systems is different. While read speech corpora have been used for Finnish, Estonian and Turkish, spontaneous telephone conversations, which are characterized by disfluencies and by the presence of non-speech have been used for Arabic experiment. This might also explain the poor performance of the Arabic system.

El-Desoky et al. [2009] investigated the use of morphological decomposition and diacritization for improving Arabic LVCSR. The authors tried to address the OOV

and short-vowel problems by using morphological decomposition and diacritization in Arabic language modeling. In their study, the vocabularies are selected from a text corpus consisting of around 206 Million full-word forms using a maximum likelihood approach where the OOV rate is minimized over held-out data. The vocabulary of the baseline word-based recognition system is 256k. The tool used for morphological decomposition and diacritization is MADA, which stands for Morphological Analysis and Disambiguation for Arabic. They did two sets of experiments: first using only morphological decomposition in language modeling and second using both morphological decomposition and diacritization. In both cases, however, the vocabulary is not purely morph vocabulary. Instead, the first N highly ranked (according to the maximum likelihood vocabulary selection procedure) decomposable words are left unsegmented. They experimented using different values for N. The best result (0.5 absolute word error rate reduction over the baseline system) has been obtained with a system that uses 256k vocabulary (236k morph and 20k full-word) and only morphological decomposition in language modeling .

Heintz [2010] did a study on Arabic language modeling with stem-derived morphemes for automatic speech recognition. To decompose words into morphemes, she used an algorithm which first identifies the stem of a word (based on stem patterns) and considers any letters on either side of the word as affixes. The claim of using this decomposition method is that besides solving the OOV problem, it also solves the dialectal and short-vowel problems of the language. The Modern Standard Arabic portion of the TDT4 multilingual broadcast news speech text and annotations, distributed by the linguistic data consortium, has been used for training and testing language models. Of this text, 14 million words are used for training while 17k and 19k words are used for development and test sets, respectively. SRILM has been used for language model training. As the author indicated, although the morpheme-based models showed improvement in terms of coverage and average negative log-probability (a metric used in Kirchhoff et al. [2002], Siivola et al. [2003] and Kirchhoff et al. [2006]), no improvement has been obtained in word recognition accuracy. She attributed this to a number of factors including acoustic confusability and lack of context in the predictions of morpheme-based models.

Using sub-word units in both acoustic and language models of a speech recognition system is a solution for the out-of-vocabulary problem. However, often these units are acoustically highly confusable and their use reduces the scope of the language model. Kiecza and Waibel [1999], while developing a LVCSR system for Korean, tried to overcome these problems by creating a set of units that lie between longer units (word phrases - "eojeols") and shorter units (syllables). They start from the syllable based system and repeatedly concatenate the syllables in

order to decrease acoustic confusability and increase the span of the n-gram language model. Similarly, Kwon [2000] developed LVCSR systems for Korean using syllables and morphemes as recognition units. Kwon [2000] tackled the problems of acoustic confusability and the limited scope of the n-gram language models by combining recognition units. Both linguistic and statistical methods have been used to concatenate units.

Choueiter et al. [2006] developed morpheme-based LVCSR system for Arabic. Unlike Kiecza and Waibel [1999] and Kwon [2000], here a morpheme lattice constrainer has been used to reduce the decoding of illegal morpheme sequences which results in non-word output and consequently lead to performance degradation. The lattice constrainer is a finite state acceptor that allows only legal sequences of morphemes. Although this method helps to avoid the recognition of morpheme sequences that lead to non-word units, it does not really solve the problem of acoustic confusability which results from the use of short units.

2.6.3 Factored Language Model

Factored language models (FLM) have first been introduced by Kirchhoff et al. [2002] for combining various morphological information in Arabic language modeling. In FLM a word is viewed as a bundle or vector of K parallel factors, that is, $w_n \equiv f_n^1, f_n^2, ..., f_n^k$. The factors of a given word can be the word itself, stem, root, pattern, morphological classes, or any other linguistic element into which a word can be decomposed. The idea is that some of the feature bundles (for example: roots, patterns and morphological class) can uniquely define the words i.e. $(W = w_i) \equiv (R = r_i, P = p_i, M = m_i)$. Therefore, the n-gram probabilities can be defined on the basis of these features/factors as follows:

$$\begin{aligned} & P(w_i|w_{i-1}, w_{i-2}) \\ & \quad = P(r_i, p_i, m_i|r_{i-1}, p_{i-1}, m_{i-1}, r_{i-2}, p_{i-2}, m_{i-2}) \\ & \quad = P(r_i|p_i, m_i, r_{i-1}, p_{i-1}, m_{i-1}, r_{i-2}, p_{i-2}, m_{i-2}) \\ & \qquad P(p_i|m_i, r_{i-1}, p_{i-1}, m_{i-1}, r_{i-2}, p_{i-2}, m_{i-2}) \\ & \qquad\quad P(m_i|r_{i-1}, p_{i-1}, m_{i-1}, r_{i-2}, p_{i-2}, m_{i-2}) \end{aligned} \tag{2.34}$$

There are two important points in the development of FLM: choosing the appropriate factors which can be done based on linguistic knowledge or using a data driven technique and find the best statistical model over these factors. Unlike normal word or morpheme-based language models, in FLM there is no obvious natural backoff

order. In a tri-gram word based model, for instance, we backoff to a bi-gram if a particular tri-gram sequence is not observed in our corpus by dropping the most distant neighbor, and so on. However, in FLM the factors can be temporally equivalent and it is not obvious which factor to drop first during backoff. If we consider a quadro-gram FLM and if we drop one factor at a time, we can have six possible backoff paths as it is depicted in Figure 2.2 and we need to choose a path that results in a better model. Therefore, choosing a backoff path is an important decision to be taken in FLM. There are three possible ways of choosing a backoff path: 1) Considering a fixed path based on linguistic or other reasonable knowledge; 2) Generalized all-child backoff where multiple backoff paths are chosen at run time; and 3) Generalized constrained-child backoff where a subset of backoff paths is chosen at run time Kirchhoff et al. [2008]. A genetic algorithm for learning the structure of a factored language model has been developed by Duh and Kirchhoff [2004].

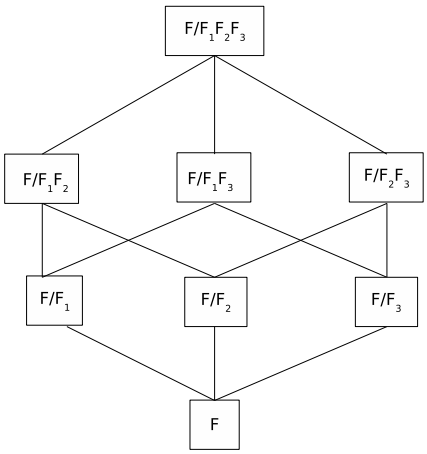

Figure 2.2: Possible backoff paths

The advantages of FLMs as indicated by Kirchhoff et al. [2003] are:

- reliable estimation of component probabilities, since more observations will be available for different combinations of morphemes,
- model simplification by avoiding superfluous conditioning variables,
- expressing dependencies across words, and
- easy integration of other word features (e.g. semantics) beyond morphological features.

Chapter 3

Amharic and Its Morphology

3.1 Introduction

Amharic is a member of the Ethio-Semitic languages, which belong to the Semitic branch of the Afroasiatic super family [Voigt, 1987]. It is related to Hebrew, Arabic, and Syrian. Amharic, which is spoken mainly in Ethiopia, is the second populous Semitic language, after Arabic. According to the 1998 census, it is spoken by over 17 million people as a first language and by over 5 million as a second language throughout different regions of Ethiopia. For fifty years, it was the constitutionally recognized national language of Ethiopia, the required language of instruction in the primary school (1-6 grades) and a required subject of the Ethiopian School Leaving Certificate Examination (ESLCE) [Anbessa and Hudson, 2007]. Currently, Amharic is the official working language of the federal democratic republic of Ethiopia and several of the states within the federal system. The language is also spoken in other countries such as Egypt and Israel [Ethnologue, 2004].

Like other Semitic languages such as Arabic, Amharic exhibits a root-pattern morphological phenomenon. In addition, it uses different affixes to create inflectional and derivational word forms. The following is a description of the Amharic word morphology.

3.2 Root-Pattern Morphology

Semitic languages are characterized by a root-pattern morphology. A root, also called radical, is a set of consonants (commonly three, but ranges from one to six) which carry the basic lexical meaning. A pattern, also called vocalic element, consists of vowels which are inserted (intercalated) among the consonants of the root. The pattern is combined with a particular prefix or suffix to create a single grammatical form [Bender et al., 1976] or another stem [Baye, 2000EC]. Stems are, therefore, formed by intercalating the vowels among root consonants. Finally, prefixes and suffixes are added either to complete the stem as a word or to form another stem.

Like other Semitic languages, Amharic is characterized by the root-pattern non-concatenative morphology. For example[1], the Amharic root ሰብር /sbr/[2] means 'break', when we insert the pattern ä-ä among the radicals, we get the stem ሰበር- /säbbär-/. Attaching the suffix -ä gives ሰበረ /säbbärä/ 'he broke' which is the first form of the verb (third person masculine singular in past tense as in other Semitic languages) [Bender et al., 1976]. Using the same pattern, but without geminating the second consonant and attaching the suffix -a, we get the process nominal ሰበራ /säbära/ which means 'breaking'. The pattern ä-a combined with the suffix -i makes agent nouns, as in ሰባሪ /säbari/ 'one who breaks'. The same stem with the suffix -a, gives us the participle ሰባራ /säbara/ 'broken'. Intercalating the vowel ä between the second and third consonant results in a jussive stem ስበር- /sbär-/. Attaching the prefix መ- /mä-/ to it forms an infinitive verb መስበር /mäsbär/ 'to break' [Bender et al., 1976]. The imperfective stem is derived from the root by intercalating the vowel ä between the first and the second consonants. Usually the form taken by the stems of a particular class is indicated by a sequence of Cs and Vs, called template. For example, CVCCVC, CCVC and CVCC are templates that represent the perfective, jussive and imperfective stems of tri-radical verbs, respectively.

3.3 Derivational and Inflectional Morphology

There are five parts of speech in Amharic: adjectives, nouns, verbs, adverbs, and prepositions and conjunctions. Prepositions and conjunctions are totally unproductive. Adverbs are few in number and are less productive. They are not inflected but some adverbs can be derived from adjectives, for instance, ክፋኛ /kIfuña/ 'severely or seriously' is derived from the adjective ክፋ /kIfu/ 'wicked' by suffixing -ña. Therefore, our discussion of derivational and inflectional morphology concentrates on the remaining three parts of speech, namely verbs, nouns, and adjectives.

3.3.1 Derivation

3.3.1.1 Verbs

Unlike the other word categories such as nouns and adjectives, the derivation of verbs from other parts of speech is not common. Anbessa and Hudson [2007] indicated the existence of verbs, called denominals, which are derived from nouns by taking just the consonants of a noun and considering them as a root. For instance, the verb መረዘ

[1] Most of the examples given in this chapter are taken from text books.

[2] For transcription purpose, IPA representation is used with some modifications. The IPA representation (with the modification we made) is given in Appendix A.

/märräzä/ 'He poisoned' and ተረተ /tärrätä/ 'He told a story' are denominal verbs derived from the nouns መርዝ /märz/ 'poison' and ተረት /tärät/ 'story', respectively. However, the authors themselves indicated that the history of denominals is rarely certain. In addition, even if the denominals are considered to be derived from the noun, we can still see that the verb derivation is from the consonants of the nouns. Thus, it is possible to say that in almost all the cases Amharic verbs are derived from root consonants, which carry the basic lexical meaning, by intercalating vowel patterns and/or using different derivational morphemes.

Roots and patterns

Most scholars (to mention some: Dawkin [1960], Bender and Hailu [1978], Leslau [2000]) claim that the number of consonants of roots in Amharic verbs range from one to six. For instance, Leslau [2000] classified the verbs as bi-radicals (abbreviated tri-radicals), tri-radicals, quadri-radicals, abbreviated quadri-radicals and pluri-radicals (those consisting of more than four consonants) on the basis of the number of consonants that show up in the surface forms of the verbs. Dawkin [1960] classified verbs into five groups depending on the number and behavior of the consonants in the verbs:

- Group I — Uncontracted three-radical verbs
- Group II — Contracted three-radical verbs with a vowel instead of the last radical
- Group III — Contracted three-radical verbs with a vowel instead of the penultimate radical
- Group IV — Uncontracted four-radical verbs
- Group V — Contracted four-radical verbs with a vowel instead of the last radical

However, Baye [1999] argues that Amharic verbal stems have uniformly three radicals and that variation in the number of consonants in the surface forms is a result of reduction and/or extension of one or more of the three radicals. That means verbs having less than three consonants in their surface form have gone through the process of root-reduction and those that have more than three passed through the process of root-extension. Here, it is important to note that scholars who claim that the number of root consonants in Amharic verbs range from one to six do not deny the existence of root reduction processes in Amharic verbs. The reduction process

involves reduction of the laryngeals and glides (h, y, w, ?) and two other weak radicals (b and r). The extension process is divided into: Internal extension (gemination and reduplication) and external extension (epenthesis). But all the scholars agree that tri-radical verbs are the basic/common types in Amharic as it is true for other Semitic languages.

Discussing whether Amharic verbs consist of uniformly three, less or more than three consonants is not the purpose of the present work. The most interesting point for us is that Amharic verbal stems are derived from root consonants by interdigitating vowel patterns. However, as we manually segmented Amharic words using the list of roots in Bender and Hailu [1978] to cross check the roots that we found during the segmentation task, we followed the long standing claim that Amharic verbs can have one to six consonants in the root.

The penultimate radical is the most important element in a verb. Specifically, "it is like the pivot of the verbal stem" [Dawkin, 1960]. Thus, traditionally Amharic verbs are classified into three types depending on the gemination pattern of this radical. In type A verbs, the penultimate radical geminates in perfect tense only. In type B verbs, the penultimate radical geminates irrespective of the verb forms. In type C verbs, the penultimate radical geminates in both perfect and imperfect verb forms.

The vowel /ä/, also called the thematic or aspectual vowel, is considered to be the only vowel that is intercalated among the consonants of all Amharic verbs except Type C verbs that are also characterized by using the vowel /a/ after the first radical. Thus, any other vowel in the surface form of Amharic verbs indicate reduction of laryngeals, sometimes called gutturals or weak radicals, and glides (h, y, w, ?). The presence of the vowel /a/ in surface forms implies the reduction of the two consonants /?/ and /h/, the vowel /o/ implies the reduction of /w/ and /e/ implies the reduction of the consonant /y/, as shown in Table 3.1.

Roots	Verbs	Gloss
frh	färra	'he feared'
n^wr	norrä	'he lived, it existed'
h^yd	hedä	'he went'
l?k	lakä	'he delegated'

Table 3.1: Reduction of consonants

Baye [1999] indicated the existence of two more weak radicals (/r/ and /b/), that are not laryngeals but sometimes disappear from the surface forms of verbs. For instance, the verbal forms አይ- /ayy/ 'saw' and አል- /all/ 'said', derived from the

tri-radical roots /r-?-y/ and /b-h-l/, respectively. Thus, the existence of the above mentioned six consonants (w, y, h, ?, b and r) in verbal roots imply the existence of a vowel other than /ä/ and the change of the vowel caused by the loss of the radicals. We considered this fact in the morphological segmentation of Amharic words.

Simple and derived verbs

Although almost all Amharic verbs are derived from root consonants, as indicated by Sisay [2004] traditionally a distinction is made between simple and derived verbs. **Simple verbs** are those verbs derived from roots by intercalating vowel patterns whereas **derived verbs** are considered as derivatives of simple verbs. The derivation process can be an internal one in which consonant-vowel patterns are changed, an external one where derivational affixes are attached to the simple derived verbs or a combination of the internal and external derivational processes.

Simple stems are formed by intercalating the vowel ኧ /ä/ (except for type C verbs) among the root consonants. The number of vowels required differs according to the conjugation and the number of consonants in the root. Table 3.2 indicates, as an example, how the pattern differs depending on conjugation and verb type for tri-radical verbs.

Verb forms	Type A: sbr		Type B: flg		Type C: mrk	
	stems	Template	stems	Template	stems	Template
Perfect	säbbär-	CVCCVC	fälläg-	CVCCVC	marräk-	CVCCVC
Imperfect	säbr-	CVCC	fällg-	CVCCC	marrk-	CVCCC
Jussive	sbär-	CCVC	fällg-	CVCCC	mark-	CVCC
Gerund	säbr-	CVCC	fällg-	CVCCC	mark-	CVCC
Infinitive	sbär-	CCVC	fälläg-	CVCCVC	maräk-	CVCVC

Table 3.2: Simple verb conjugation

The following are stems derived from the simple stems.

1. **Causative:** Causative verbs are derived by adding the derivational morphemes a- and/or as- to the verb stem as in the examples ደረስ- /därräs-/ 'arrive' - አደረስ- /adärräs-/ 'cause to arrive' and ወሰድ- /wässäd-/ 'take' - አስወሰድ- /aswässäd-/ 'cause to take'. In most cases the a- morpheme is used to form causative of intransitive verbs and verbs of state, and the as- for transitive ones. Exceptions are:

   - verbs that begin with a, which always take the morpheme as- to form causative e.g. አደገ /addägä/ 'he grew' - አሳደገ /asaddägä/ 'he raised';

- transitive verbs which have to do with eating, drinking and related activities whose causative is formed with the a- morpheme as in በላ /bälla/ 'eat' - አበላ /abälla/ 'feed';
- verbs which are transitive only in the sense of having cognate objects form their causatives using the morpheme as- as in ጨፈረ /Čäffärä/ 'he danced' - አስጨፈረ /asČäffärä/ 'he caused to dance'

There are verbs that take both the a- and as- causative morphemes to form the direct causative and indirect causative, respectively. For instance, the verb መጣ /mät't'a/ 'he came' can take both derivational morphemes as in አመጣ /amät't'a/ 'he brought' and አስመጣ /asmät't'a/ 'he caused someone to bring something'. In direct causative, the causer actively participate in action, but in indirect causative the causer gets the action performed by somebody else.

2. **Passive/Reflexive:** The passive verbs are derived using the derivational morpheme ት(ኧ)- /t(ä)-/. This derivational morpheme is realized as ተ- /tä-/ before consonants and as ት- /t-/ before vowels. Moreover, in the imperfect, jussive and in derived nominals like verbal noun, the derivational morpheme ት- /t-/ is used. In this case, it assimilates to the first consonant of the verb stem, and as a result, the first radical of the verb geminates [Mengistu, 2002]. For example, in ይ-ወርወር /y-wwärwär/ 'let it be thrown' which is the surface form of /y-t-wärwär/, the gemination of the first consonant, namely /w/, is the result of assimilation of the passive prefix ት- /t-/.

   In Amharic, the passive does not normally apply to intransitive verbs. Exceptions are intransitive verbs like ፈላ /fälla/ 'it boiled' that form their passive forms using the prefix ት(ኧ)- /t(ä)-/ as in ተፈላ /täfälla/ 'it was boiled'. However, some scholars [Bender and Hailu, 1978, Anbessa and Hudson, 2007] argue that such kind of verbs derive their passive from their causative form (አፈላ /afäll'a/ 'he boiled') although they give no explanation about how the causative prefix disappears from the surface form of the passive verb. Others, like [Demoz, 1964] argue that most intransitive verbs, which allow the passive, take a special type of object – a cognate object – and, therefore, the attachment of the passive prefix is not problematic since such constructions are considered transitive.

   Intransitive verbs may take the morpheme t(ä)- to express the generalized/habitual impersonal [Anbessa and Hudson, 2007] and the irony or sarcasm meanings [Mengistu, 2002].

   The reflexive form of self-grooming transitive verbs such as 'wash' and 'shave' is

derived by attaching the prefix ተ- /tä-/ to them. For example, አጠበ /at't'äbä/ 'he washed' - ታጠበ /tat't'äbä/ 'he washed himself'. Since the same prefix is also used to render a passive reading, the reflexive verb can also have a passive interpretation. However, Mengistu [2002] said that with verbs that express events that normally affect a body part, reflexive is the preferred reading. Although, the reflexive derivation applies to transitive verbs, it does not apply to all transitive verbs. For transitive verbs for which reflexive derivation using the prefix ተ- /tä-/ is not possible, the reflexive construction can be formed by employing the nominal reflexive strategy. For instance, ተመታ /tämätta/ which is derived from መታ /mätta/ has passive meaning 'he was hit' instead of reflexive. The reflexive is, therefore, given using nominal reflexive strategy as እራሱን መታ /?rasun mätta/ 'he hit himself'.

The intransitive/anticausative verb is also derived from the transitive ones using the prefix /tä-/. Thus, verbs like ተሰበረ /täsäbbärä/ can have either a passive 'be broken' or an anticausative reading 'break, intransitive'.

3. **Reduplicative/repetitive:** Reduplicative stems indicate an action which is performed repeatedly. For tri-radical verbs, such stems are formed by duplicating the second consonant of the root and using the vowel ኣ- /a-/ after the duplicated consonant as in ሰባበረ /säbabärä/ 'he broke repeatedely'. All verb types, Type A, B and C have the same reduplicative forms.

4. **Reciprocal:** Reciprocal verbs are derived by prefixing the derivational morpheme ተ- /tä-/ either to the derived type C forms (that use the vowel a after the first radical) or to the reduplicative stem [Mengistu, 2002]. For example, reciprocal forms of ተጋደሉ /tägaddälu/ 'killed each other' and ተገዳደሉ /tägädaddälu/ 'killed one another' are derived from the derived type C stem gaddäl- and reduplicative stem gädaddäl-, respectively. Although Mengistu [2002] indicated the fact that no essential semantic difference exists between the two reciprocal forms, Baye [1999] said that there is difference in the number of participants in the reciprocal action. In the case of a reciprocal that is derived from a derived type C stem, there are two participants, whereas in the form derived from a reduplicative stem, the number of participants may be more than two.

The causative of reciprocal verbs are formed by adding the causative prefix a- to the reciprocal verb forms. However, the reciprocal prefix t(ä)- assimilates to the stem-initial consonant (thus causes the first radical of the stem to geminate) and does not show up in the surface form of the reciprocal causative.

That means, for example, the form, አ-ት-ጋደል- /a-t-gaddäl-/ is changed to አ-ጋደል- /a-ggaddäl-/ 'caused to kill each other'.

5. **Adjutative:** prefixing the causative prefix a- to a stem which is formed by geminating the first consonant and using the vowels ኣ /a/ and ኧ /ä/ after the first and second consonants, respectively results in an adjutative stem [Baye, 1999]. Although, the verbs are derived using the causative prefix, they indicate a participatory or adjutative subject. The surface form of the adjutative verbs is similar to the reciprocal causative ones, and therefore, they are often confused. However, unlike the reciprocal causative verb forms, the gemination of the first radical in adjutative verbs is not a result of assimilation. For example, the adjutative form አ-ጋደል- /a-ggaddäl-/ (compare the form with the reciprocal causative) means 'help somebody to kill somebody else'.

6. **Other derived verb forms:** Attenuative and intensive verb stems are the other two kinds of stems derived from the verbal roots. These stems are bound stems as they always require the auxiliary verb /allä/ (for intransitive verbs) or /adärrägä/ (for transitive verbs) and verbal features like tense and aspect and nominal features like person, number and gender are expressed using these auxiliary verbs. Consequently, Baye [1999] characterized the attenuative and the intensive stems as non-verbal stems. An attenuative stem is derived with the intercalation of the vowel /ä/ and gemination of the ultimate radical. This stem, as its name implies, shows attenuated actions. For example, ከፊት አለ /käfätt allä/ expresses the attenuation action 'it opened slightly'. On the other hand, the intensive stem is a bound stem which is formed by extending both the ultimate and penultimate radicals and inserting the epenthetic vowel /I/ to avoid consonant clusters. ክፍት አለ /kIffItt allä/ means 'it opened suddenly'. In addition to these stems, the reduplicative form of a verb can also express intensive and attenuated actions [Leslau, 2000]. For instance, the duplicative verbs ሰባበረው /säbabäräw/ and ቀማመሰ /k'ämamäsä/ in ጥይቱ መስታወቱን ሰባበረው /t'ytu mästawätun säbabäräw/ 'The bullet shattered the glass' and ጥቂት ምግብ ቀማመሰ /t'k'it mIgb k'ämamäsä/ 'he ate some food' show the intensive and attenuated action, respectively.

3.3.1.2 Nouns

The nouns in Amharic include notional words denoting subjects, objects, phenomena; and also words used as objects of thought, any actions and states, features and relations. Whether a word is a noun or not can be determined according to morphological distinction and sometimes syntactically [Titov, 1976]. A word is grouped

under noun if it inflects for the Amharic plural marker ‑ኦች /-očč/, can be used as a subject or an object in a sentence, is modified by adjectives, and comes after demonstrative pronouns.

Amharic nouns can be either primary or derived. They are derived if they are related in their root consonants and/or meaning to verbs, adjectives, or other nouns. Otherwise, they are primary. For example, a noun እግር /?Igr/ 'foot, leg' is primary but, እግረኛ /?Igräññña/ 'pedestrian' is derived from the nominal base እግር by adding the morpheme ‑ኛ /-äññña/ [Leslau, 2000]. The following is a description of noun derivation.

Nouns are derived from other nouns, adjectives, roots, stems, and the infinitive form of a verb by affixation and intercalation. The morphemes ‑ነት /-nät/, ‑ኛ /-äññña/, ‑ኧት /-ät/, ‑አዊ /-awi/, ‑ተኛ /-täññña/, ‑ኛ /-ña/ and the prefix ባለ- /balä-/ are used to derive nouns from other nouns. Table 3.3 shows examples of nouns derived from other basic nouns.

Base noun	Gloss	Bound morpheme	Derived noun	Gloss
lIğ	child	-nät	lIğnät	childhood
bärr	door	-äññña	bärräññña	goal keeper
šum	appointed	-ät	šumät	appointment
ğärmän	Germany	-awi	ğärmänawi	German
dInbär	border	-täññña	dInbärtäññña	one who shares a border
?ngliz	England	-ña	?nglizña	English
mäkina	car	balä-	balämäkina	car owner

Table 3.3: Nouns derived from other nouns

From the adjectives, nouns can be derived using the suffixes /nät/ and /-ät/ as in the examples /dägnät/ 'generosity' which is derived from the adjective /däg/ 'kind' and Iwqät 'knowledge' from the adjective Iwq 'known'.

Nouns can also be derived from verbal roots by intercalation and affixation. Table 3.4 shows nouns derived from a verbal root, their category, the vocalic pattern and the affixes used. As it can be seen from the table, one possibility to derive a noun from a root is intercalating the vowel /I/[3] among the root consonants or just after the first root consonant. This intercalation may also result in a bound morpheme which, together with different affixes, form other nouns. Most of the nouns formed this way are resultative nouns although sometimes they are also process nouns. Nouns like ዝናብ /znab/ 'rain' or bound stems such as ትካዝ- /tIkkaz-/ and ውዳቅ-

[3]As Baye [2000EC] indicated the insertion of /I/ is necessary for purely phonological reasons. It is an epenthetic vowel inserted to avoid consonant cluster.

/wIddak'-/, which are used to derive nouns, can be derived by intercalating the vowel ኣ /a/ after the second consonant of a root. Intercalation of the vowel ኧ /ä/ after the first radical or among radicals results in either a noun or a bound stem used to derive a noun. The pattern ä-a and the suffix /i/ are used in the derivation of agent nouns. Nouns of manner can be derived by prefixing /a/ to the stem which is formed by duplicating the penultimate radical and intercalating the pattern ä-a-ä. The infinitive/verbal noun is derived by prefixing the morpheme /mä-/ to the jussive verb stem and the instrumental noun is derived by suffixing /-iya/ to the infinitive.

Root	Stem	Affix	Derived noun	Category	Gloss
l-b-s	lIbs		lIbs		cloth
t'-k'-m	t'Ik'm		t'Ik'm		advantage
g-r-d	gIrd	-oš	gIrdoš		awning, eclipse
g-b-r	gIbr	-nna	gIbrInna		farming, agriculture
s-n-f	sInf	-nna	sInfInna		laziness
d-g-m	dIggIm	-oš	dIggImoš	process/ resultative	repetetion
s-b-r	sIbr	-at	sIbrat	resultative	breakage
s-r-k'	sIrk'	-ot	sIrk'ot		theft
č-h-l	čIl	-ota	čIlota	resultative	ability
d-k-m	dIkam		dIkam	resultative	tiredness
z-n-b	zInab		zInab	resultative	rain
t-k-z	tIkkaz	-e	tIkkaze	resultative	melancholy, sadness
w-t'-h	wIt'	-et	wIt'et	resultative	result
g-f-h	gIf	-it	gIfit	resultative	influence
s-r-Č	sIrČ	-it	sIrČt	process	transmission
g-f-h	gIf	-iya	gIfiya	process	crush
š-f-t	šIft	-a	šIfta	resultative	outlaw, bandit
m-l-s	mäls		mäls	resultative	answer
k'-l-d	k'äld		k'äld		'joke'
k'-l-m	k'äläm		k'äläm	resultative	color, ink
s-b-r	säbär-	-a	säbära	process	process of breaking
f-t-n	fätän-	-a	fätäna	resultative	test ,exam
s-b-k	säbak-	-i	säbaki	agent	preacher
w-d-k'	wIddak'-	-i	wIddak'i	resultative	reprobate, rubbish
s-b-r	sbär	mä-	mäsbär	infinitive	to break
f-l-g	fälläg	mä-	mäfälläg	infinitive	to seek
s-b-r	mäsbär	-iya	mäsbäriya	instrumental	tool for breaking
s-b-r	ssäbabär	a-	assäbabär	manner	manner of breaking

Table 3.4: Nouns derived from verbal roots

In Amharic, nouns can also be formed through compounding. For example,

እንጀራእናት /?nIğära?nat/ 'stepmother' is derived from the nouns እንጀራ /?nIğära/ 'Ethiopian bread' and እናት /?nat/ 'mother'. As it can be seen, no morpheme is used to bind the two nouns. But, there are also compound nouns whose components came together by inserting the compounding morpheme ኧ /ä/ as in ቤተክር-ስቲያን /betäkrIstiyan/ 'church' which is formed from ቤት /bet/ 'house' and ክርስቲያን /krIstiyan/ 'Christian'.

3.3.1.3 Adjectives

Adjectives in Amharic include all the words that modify nouns and can be modified by the word በጣም /bät'am/ 'very, greatly'. As it is true for nouns, adjectives can also be primary (such as ደግ /däg/ 'kind') or derived, although the number of primary adjectives is very small.

Adjectives are derived from nouns, stems or verbal roots by adding a suffix and by intercalation. The suffixes -ኣም /-am/, -ኛኛ /-äññä/, -ኣዊ /-awi/, and -ኣማ /-ama/ are used in the derivation of adjectives from nouns. For example, it is possible to derive ሀብታም /habtam/ 'rich, wealthy', ሀይለኛ /hayläña/ 'powerful, mighty', ዘመናዊ /zämänawi/ 'modern' and ድንጋያማ /dIngayama/ 'stony' from the noun ሀብት /habt/ 'riches, wealth', ሀይል /hayl/ 'power, force', ዘመን /zämän/ 'period, epoch' and ድንጋይ /dInIgay/ 'stone', respectively.

Adjectives can also be derived either from roots by intercalation of vocalic elements or attaching a suffix to bound stems. Table 3.5 gives examples of adjectives derived from bound stems or verbal roots. As it can be seen from the table, the suffix -a is used to derive adjectives from a bound stem which is formed by intercalating the vocalic element ä-a to the root consonants. The vocalic patterns ä-a, ä-ä (also used in the derivation of verbs and nouns), ä-i, and u are used to derive adjectives from roots.

Root	Stem	Affix	Derived Adjective	Gloss
t'-m-m	t'ämam-	-a	t'ämama	crooked, bent
k-b-d	käbad		käbad	heavy
s-n-f	sänäf		sänäf	lazy
r-z-m	räzzim		räžim	long
k-b-r	k-bur		kIbur	respectful
z-n-g-h	z-n-gu		zIngu	forgetfull

Table 3.5: Adjectives derived from verbal roots

Adjectives can also be formed through compounding. For instance, ሆደሰፊ /hodäsäfi/ 'tolerant, patient', is derived by compounding the noun ሆድ /hod/ 'stom-

ach' and the adjective ሰፊ /ssäfi/ 'wide'. Note the use of ኧ /ä/ as a compounding morpheme.

3.3.2 Inflection

3.3.2.1 Verbs

Verbs are inflected for person, gender, number, aspect, tense, and mood [Baye, 2000EC]. Table 3.6 shows the inflection of perfective and imperfective verbs for person, gender, number.

Person	Perfective	Imperfective
1^{st}	säbbär-ku/hu	?-säbr
1^{st} plural	säbbär-n	?-n-säbr
2^{nd}		
masculine	säbbär-h/k	t-säbr
feminine	säbbär-š	t-säbr-i
polite	säbbär-u	t-säbr-u
plural	säbbär-ačču	t-säbr-u
3^{rd}		
masculine	säbbär-ä	y-säbr
feminine	säbbär-äčč	t-säbr
polite	säbbär-u	y-säbr-u
plural	säbbär-u	y-säbr-u

Table 3.6: Subject markers

As it can be seen from the table, in imperfective stems the morphemes አ- /?-/, ት- /t-/ and ይ- /j-/ indicate first, second and third person, respectively. -ኢ /-?i/ is the gender marker suffix and ን- /n-/ and -ኡ /-u/ are plural markers in the first and second person. For perfective verbs, the markers for the first, second and third person are -ኩ /-ku/ or -ሁ /-hu/, -ክ /-k/ or -ህ /-h/ and -ኧ /ä/, respectively. These morphemes are attached to the perfective verb stem as suffixes. The gender and number marker suffixes are assimilated with the person marker suffixes as in -ሽ/š/ which resulted from the assimilation of the second person marker -ህ /-h/ and the gender marker -ኢ /-?i/. These markers (person, gender and number) are called subject markers as they indicate or substitute the subject in a given sentence.

There are also person markers that indicate the object in a sentence. These morphemes can be seen from Table 3.7. The morpheme which comes immediately after the stem of the verb is the third person subject marker, -ኧ /ä/. After this subject morpheme, come the object morphemes. That means if both subject and object markers are attached to a verb stem, their order is always stem-subj-obj. Here

object can be either a direct or prepositional object but both object markers can not appear together on a verb. Prepositional object markers indicate prepositional phrases that have different adverbial functions. The prepositions that are used to form the prepositional phrases are ለ /lä/ and በ /bä/ that have benefactive and malfactive functions, respectively. For instance, እንጨቱን ሰበረልኝ /?nIČätun säbbärällň/ means 'he broke the wood for me' whereas እንጨቱን ሰበረብኝ /?nIČätun säbbäräbbň/ means 'He broke the wood - against my will'.

Person	Direct object	Prepositional objects	
		Benefactive	Malfactive
1^{st}	säbbär-ä-ň	säbbär-ä-ll-ň	säbbär-ä-bb-ň
1^{st} plural	säbbär-ä-n	säbbär-ä-ll-n	säbbär-ä-bb-n
2^{nd}			
masculine	säbbär-ä-h	säbbär-ä-ll-h	säbbär-ä-bb-h
feminine	säbbär-ä-š	säbbär-ä-ll-š	säbbär-ä-bb-š
polite	säbbär-ä-wo(t)	säbbär-ä-ll-wo(t)	säbbär-ä-bb-wo(t)
plural	säbbär-ä-ačču	säbbär-ä-ll-ačču	säbbär-ä-bb-ačču
3^{rd}			
masculine	säbbär-ä-w	säbbär-ä-ll-ät	säbbär-ä-bb-ät
feminine	säbbär-ä-at	säbbär-ä-ll-at	säbbär-ä-bb-at
polite	säbbär-ä-ačč̈aw	säbbär-ä-ll-ačč̈aw	säbbär-ä-bb-ačč̈aw
plural	säbbär-ä-ačč̈aw	säbbär-ä-ll-ačč̈aw	säbbär-ä-bb-ačč̈aw

Table 3.7: Object markers

With regard to aspect, Amharic verbs are basically categorized into two classes: perfect and imperfect forms. Under these main categories, Baye [2000EC] and Baye [2006] identified four sub-aspectual types, three derived from the imperfective (prospective, inceptive and completive) and one derived from the perfective stem (progressive). The prospective indicates an imminent or intended action while the inceptive indicates an action which is beginning. As the names imply, progressive and completive verbs denote an action in progress and a completed action, respectively. Amharic verbs inflect for these aspects except for inceptive. The prospective aspect is indicated by the prefix ል- /l-/ which comes before the subject marker of an imperfective stem which is followed by the auxiliary verbs ነው /näw/ 'is' and ነበር /näbbär/ 'was'. As it has already been said, verbs do not inflect for inceptive. It is expressed by an imperfective verb followed by the auxiliary verb, ጀመር /ğämmär/ 'begin'. Progressive is expressed using the prefix እየ- /?yyä-/ attached to a perfective stem and with the auxiliary verbs used in prospective aspect. The completive aspect (whose stem is also identified as gerund or converb) is indicated by adding different genitive suffixes for first, second and third persons and using the auxiliaries

አል- /all-/ 'exist' and ነበር /näbbär/ 'was'.

There are four moods in Amharic: declarative, interrogative, negative and imperative. Verbs can take different forms according to the mood. This form can be expressed either in the stem or by inflectional affixes. Table 3.8 shows the structure and inflection of verbs according to the mood.

Person	Declarative	Interrogative	Negative	Imperative
1^{st}	?-säbr-	l-?-sbär-	all-sbär-	
1^{st} plural	?-n-säbr-	?-n-sbär-	all-n-sbär-	
2^{nd}				
masculine	t-säbr-	t-säbr-	all-t-sbär-	sbär
feminine	t-säbr-?i	t-säbr-?i	all-t-sbär-?i	sbär-?i
polite	t-säbr-u-	t-säbr-u	all-t-sbär-u	sbär-u
plural	t-säbr-u-	t-säbr-u	all-t-sbär-u	sbär-u
3^{rd}				
masculine	y-säbr-	y-sbär-	all-y-sbär-	y-sbär
feminine	t-säbr-	t-sbär-	all-t-sbär-	t-sbär
polite	y-säbr-u-	y-sbär-u	all-y-sbär-u	y-sbär-u
plural	y-säbr-u-	y-sbär-u	all-y-sbär-u	y-sbär-u

Table 3.8: Inflection according to mood

Tense in Amharic can be broadly categorized as past and non-past. The past tense can be categorized into three: simple, recent and remote past. Simple past, that uses a perfective stem, indicates that an action is completed in the past but does not clearly indicate the time unless adverbs of time (such as ትናንት /tInant/ 'yesterday', ዛሬ /zare/ 'today' and አሁን /?ahun/ 'now') are used. For example, the sentence ካሳ አንበሳ ገደለ /Kasa ?nbässa gäddälä/ 'Kasa killed a lion' does not indicate when Kasa did the action whereas ካሳ ትናንት አንበሳ ገደለ /Kasa tInant ?nbässa gäddälä/ 'Kasa killed a lion yesterday' clearly indicates the time. Recent past is formed with a completive aspect stem and the auxiliary verb አል- /all-/ 'exist', which is attached to the verb, as in ካሳ አንበሳ ገድሏል /Kasa ?nbässa gädlo?ll/ 'Kasa killed a lion'. However, when the completive aspect is used with the auxiliary ነበር /näbbär/ 'was', it gives a remote past tense. In this case, however, the auxiliary is not attached to the completive stem as in ካሳ አንበሳ ገድሎ ነበር /Kasa ?nbässa gädlo näbbär/ 'Kasa killed a lion'.

Non-past tense is formed with the imperfective stem and the auxiliary verb አል- /all-/ 'exist', which is a bound morpheme attached to the verb. The stem indicates that the action is not performed and the auxiliary indicates when the action will be performed. This time ranges from present to future thus, the tense can be either present or future. This may be the reason why some call this tense as present-future.

Normally, this tense is ambiguous and it is difficult to know whether it is present or future unless adverbs of time are used. In addition, there is also continuous (present or past) tense in Amharic. The continuous tense is formed by attaching the prefix እየ- /?yä-/ to a verb and the auxiliaries ነው /näw/ 'is' and ነበር /näbbär/ 'was'. For instance, ካሳ መፅሀፍ እያነበበ ነው /Kasa mäs'haf ?yanäbäbä näw/ 'Kasa is reading a book' is a present continuous whereas ካሳ መፅሀፍ እያነበበ ነበር /Kasa mäs'haf ?yanäbäbä näbbär/ 'Kasa was reading a book' is past continuous tense.

3.3.2.2 Nouns

Amharic nouns inflect for case, number, definiteness, and gender marker affixes. In Amharic, there are three cases: nominative, accusative and genitive. Nominative has no indicator [Baye, 2000EC]. It is distinguished by its place in a sentence where nominative comes always before accusative [Titov, 1976], and the subject markers attached on the verb in a given sentence. The suffix -n and the suffixes shown in Table 3.9 or the prefix የ- /yä-/ inflect Amharic nouns for accusative and genitive case, respectively.

Person	Singular		Plural
	Vowel ending	Consonant ending	
1^{st}	-ዬ /-ye/	-ኤ /-e/	-አችን /-ačč n/
2^{nd}			
masculine	-ህ /-h/	-አህ /-Ih/	-አችሁ /-ačču/
feminine	-ሽ /-š/	-አሽ /-Iš/	
polite	-ዎ /-wo/	-ዎ /-wo/	
3^{rd}			
masculine	-ው /-w/	-ኡ /-u/	-አቸው /-ačč äw/
feminine	-ዋ /-wa/	-ዋ /-wa/	
polite	-አቸው /-ačč äw/	-አቸው /-ačč äw/	

Table 3.9: Genitive case markers (adapted from Titov [1976]))

In contrast to other Semitic languages, Amharic has only two numbers - singular and plural. The suffixes -ኦች /-očč/, -ኣን /-an/ and -ኣት /-at/ are used to inflect nouns for number. For example, ቤቶች /betočč/ 'houses',መምህራን /mämhIran/ 'teachers' and ህፃናት /hIs'anat/ 'babies' are plural forms of ቤት /bet/ 'house', መምህር /mämhIr/ 'teacher' and ህፃን /hIs'an/ 'baby'. Vocalic changes and reduplication are also used to indicate plurality of some words, especially words borrowed from Geez[4] language. For example, ደናግል /dänagel/ 'virgins' and ቁሳቁስ /k'usak'us/ 'things' are

[4]Geez is another Ethiopian Semitic language nowadays mostly used in Ethiopian orthodox church.

plural forms of ድንግል /dIngIl/ 'virgin' and ቁስ /k'us/ 'thing' formed through vocalic changes and reduplication.

A noun in Amharic can be either definite or indefinite. Indefiniteness has no special marker. However, as Titov [1976] said, the numeral አንድ /and/ 'one' is sometimes used to indicate indefiniteness. For instance, አንድ አልጋ /and alga/ might mean 'one bed' or 'a bed' depending on the context. The definiteness markers, which are bound morphemes, are given in Table 3.10. As it is indicated in the table, ‑ኡ /-u/ and ‑ው /-w/ are used for masculine nouns ending with consonant and vowels, respectively. For the feminine, the morpheme ‑ዋ /-wa/ is used. In addition, ‑ኢቱ /-itu/ or ‑ዩቱ /-yitu/ and ‑ኢትዋ /-itwa/ or ‑ዩትዋ /-yitwa/ can be used for feminine. The only definiteness marker for plural is ‑ኡ /-u/ which is attached to the noun after the plural marker. If a definite noun is preceded by an adjective, the definiteness marker is attached to the adjective instead of the noun. As one can observe, the definite articles of singular nouns, namely ‑ኡ /-u/, ‑ው /-w/ and ‑ዋ /-wa/, coincide with the corresponding third person genitive/possessive suffixes and are determined depending on the context. For example, ቤቱ /betu/ might mean 'his house' or 'the house'.

Gender/Number	Definiteness Markers	
	Consonant ending	Vowel ending
Sg. masc.	-u	-w
Sg. fem.	-wa, -itu, itwa	-wa, -yItu, yItwa
Pl.	-u	-u

Table 3.10: Definiteness markers (adapted from Leslau [2000]))

Like most Semitic languages, Amharic has only two genders masculine and feminine, and distinguishes gender in the second and third person. Some Amharic nouns carry gender in their meaning (for example, በሬ /bäre/ 'ox' and ላም /lam/ 'cow') whereas others require additional means to indicate gender. Masculine nouns do not have gender marker morphemes but the morphemes ኢት /-it/ and ኢቱ /-itu/[5] are used to indicate feminine nouns. Moreover, gender distinction is best observed in the gender of the definite article, the demonstrative pronouns, the verb referring to a noun or the gender specifier. For example, ሙሽራው /mušraw/ 'the bridegroom' is masculine as it is indicated by the masculine definite article ‑ው /-w/ whereas ሙሽራዋ /mušrawa/ is feminine as shown by the feminine article ‑ዋ /-wa/. The masculine demonstrative pronoun ይህ /yIh/ 'this' in ይህ ተማሪ /yIh tämari/ 'this student' indicates that ተማሪ /tämari/ 'student' is treated as a masculine in contrast

[5]The feminine markers may also be used to express diminutiveness.

to **ይህች** /yIhč/ in **ይህች ተማሪ** /yIhč tämari/ 'this(fem.) student'. The verb referring to the noun also reveals the gender of the noun as in **ፈረስ ይጋልባል** /färäs ygalbal/ 'a horse gallops' and **ፈረስ ትጋልባለች** /färäs tgalbaläč/ 'a horse(fem.) gallops'. In addition, the gender specifiers (**ወንድ** /wänd/, **ሴት** /set/, **ተባት** /täbat/, **እንስት** /?nIst/ and **አውራ** /awra/) can also indicate the gender of a noun as in **ወንድ ልጅ** /wänd lIǧ/ 'boy' and **ሴት ልጅ** /set lIǧ/ 'girl'.

3.3.2.3 Adjectives

Adjectives inflect for case, number, definiteness and gender in a similar fashion to nouns. Therefore, we do not discuss the inflection of adjectives here. However, some categories of the adjective can be marked for number through reduplication. For example, the singular form **ትልቅ** /tIlk'/ 'big', becomes **ትላልቅ** tIlAlk' in the plural.

Chapter 4

Computational Morphology

This chapter gives an introduction to computational morphology. We, specifically, present what the field is all about, its application, the output of computational morphology systems and the approaches to computational morphology. A review of some works on Amharic computational morphology are also presented.

4.1 Introduction

Morphology is the study of word formation. It tries to discover the rules that govern the formation of words from the smaller meaning bearing units, morphemes, in a language. The field that tries to perform the same task automatically using computers and computational methods is called computational morphology. It deals with the processing of words and word forms, in both their written and spoken form [Trost, 2003]. The most basic task in computational morphology is to take a string of characters or phonemes as input and deliver the analysis as output. It has a wide range of practical applications. Sproat [1992] indicated that morphological information is useful in several natural language processing areas such as text generation, parsing, lemmatization, machine translation, document retrieval, etc. Even applications that require little linguistic knowledge, e.g. information retrieval, include some amount of morphological treatment [Daille et al., 2002].

Specifically, computational morphology is applied in the following areas [Sproat, 1992]:

- **Natural language applications:** Morphological processing systems can be applied in natural language tasks such as parsing, generation, machine translation, lemmatization, and the construction and use of on-line dictionaries. In parsing, for example, one needs to know properties of words such as part-of-speech (POS) category or morphosyntactic features. These properties can be predicted, on the basis of the last suffix or the first prefix of the word, by a computational morphology system which is also called morphological analyzer or parser.

- **Speech synthesis:** Since most text-to-speech systems incorporate some amount of syntactic analysis, they use morphological information. Moreover, decomposition of morphologically complex words is important for proper pronunciation of words. For instance, decomposition of the word *boathouse* into *boat* and *house* helps the text-to-speech system not to pronounce the *th* in *boathouse* as θ (as in thing) or ð (as in father). In many cases, morphology is also responsible for word stress assignment, cf. an'alysis, anal'ytics, 'analyse.

- **Speech recognition:** Currently, speech recognition systems are based on a limited vocabulary and they do not recognize words that are not in the dictionary. However, these systems, theoretically, need to handle out-of-vocabulary words. This is why morphological analysis has become indispensable in such systems. In some systems morpheme-based recognition rather than word-based recognition is used and words are, therefore, recognized as concatenation of morphs. In such systems, the lexical and language models are also developed using morphs as units.

- **Document retrieval:** In document/text retrieval, there is a need to conflate keywords (e.g. spy and spies, church and churches, etc) in order to retrieve all documents in the database that contain those words. For morphologically impoverished languages like English, stemmers might well serve this purpose. However, finding a given word in a morphologically rich language requires a fair amount of morphological processing.

- **Word processing applications:** Word processing includes hyphenation and spelling correction. Text segmentation in some Asian languages (such as Chinese, Japanese and Korean) or Japanese text input are also part of word processing. In hyphenation, morphological analyzers can be used for segmenting words correctly. Most current spelling correction systems are based on dictionaries that list all word forms. Listing all word forms in a dictionary is, however, disadvantageous. Firstly, the dictionaries can not be complete in coverage and secondly increasing the size of the dictionary means that the program has to scan words in the dictionary which consequently makes the spelling correction system slow. Thus, it is preferable to have a root lexicon, a set of affixes and simple morphotactic rules instead of listing all word forms. In languages like Chinese, Japanese and Korean, words in a sentence are not separated by blank spaces or punctuation marks. Morphological analysis can, therefore, be used in segmentation of sentences into words. Japanese is written with a combination of two sets of characters (Kana and Kanji). As the number

of Kanji characters is very large, most Japanese text input systems use kana-kanji converter, which requires a combination of statistical and morphological methods.

4.2 Output of a Computational Morphology System

In the previous section, we have seen that morphological analyzers are important in many areas. However, different applications require different morphological information. This makes the output of a computational morphology system to depend on the application for which the system is designed [Sproat, 1992]. For example, a morphological analyzer designed to be used in a syntactic parser and another one developed for a text-to-speech application won't provide the same kind of analysis. In the former, word properties such as morphosyntactic feature are indispensable while for the latter only the sequence of morphs into which a word can be decomposed is required. On the other hand, for information retrieval systems, information that indicate the word "spies" is an inflectional form of the word "spy" is required from a morphological analyzer. Thus, having the word "spies" as an input, we can obtain at least the following three possible analyses as output (setting aside the ambiguity of spy between a noun and a verb).

I. spy + NounPlural or spy + VerbThirdPersonSingularPresentTense

II. spy + s

III. spies -> spy

Generally, "there is no hard and fast answer to the question of what kind of analysis a morphological analyzer should provide" [Sproat, 1992]. This implies that we have to design morphological analyzers in such a way that they provide the morphological information in the format required by the target system that uses the information. For the development of morpheme-based language models, morphological analyzers that provide the second analysis (from the above list) are required.

4.3 Approaches to Computational Morphology

Understanding the importance of computational morphology in various speech and language processing applications, a number of approaches are used to develop computational morphology systems. These approaches are based on concepts in automata theory, probability, the principle of analogy, or information theory [Kazakov and Manandhar, 2001]. Kazakov and Manandhar [2001] broadly categorized

the approaches of computational morphology into rule-based and corpus-based. Soudi et al. [2007] also used the same classification but with different terminologies: knowledge-based and empirical approaches instead of rule- and corpus-based, respectively.

4.3.1 Rule-based Approaches

A rule-based or knowledge-based approach is built on solid linguistic grounds. Because of their reliance on linguistic frameworks, systems developed using rule-based approaches are often efficient and produce better quality outputs [Karttunen, 1994].

The simplest model of morphology can be the one in which all words are listed along with their morphological features. Morphological analysis is, therefore, performed as a table lookup. This approach, called *full-form lexicon* by Trost [2003], is simple and applicable to all possible morphological phenomena. However, it suffers from redundancy since most natural languages exhibit at least some productive morphological processes by which a great number of word forms are created. In addition, this method does not have a means to cope with out-of-lexicon word forms.

Another approach that solves the problem of redundancy in a full-form lexicon is a lemma lexicon. Unlike the full-form lexicon, in a lemma lexicon, we store the lemma of words as representative for all the different forms of a paradigm. Then, we use an algorithm that relates every form to its lemma and also delivers a morphosyntactic analysis. If we consider concatenative morphology, for instance, affixes must be stored in a separate list together with the relevant morphotactic rules. The problem of morphological analysis can, therefore, be considered as finding a sequence of affixes and a lemma that conforms to the morphotactic. However, this approach has limitations with regard to the treatment of morphological rules. For instance, if a word does not conform to the regular default case, then we need some kind of exception handling mechanism, which can be tailored to a particular language. Moreover, the algorithm is language specific and most likely we need to develop separate algorithms for analysis and generation [Trost, 2003], hence there is a need for other ways of approaching the problem.

The most common and popular rule-based method, which is devised to handle morphological analysis and generation in a bi-directional way, is the Two-Level-morphology (TLM) [Koskenniemi, 1983]. TLM is based on two levels of representations (underlying or lexical level and surface level), and a set of morphological rules, which are compiled into finite state transducers, used to map the underlying and surface level word forms. The lexicon, that lists the lexical morphemes, can be divided into different logical sublexicons (noun stem, verb stem, suffix, etc. lexicon) each

implemented as a finite state automaton. TLM is based on the assumption that surface forms are constructed by a concatenation of lexical morphemes. Originally this assumption made the use of TLM for languages with non-linear morphology very difficult. More recently, however, it has also been applied to Semitic languages that are characterized by nonconcatenative morphology e.g. Amharic, Arabic, Akkadian and Hebrew.

A feature peculiar to rule-based approaches is that they are based on linguistic knowledge presented to the systems as linguistic resources, such as the lexicon and the morphological rules. That means such resources, particularly the rules need to be handcrafted for each language. Consequently, the development of a rule-based morphological analyzer is costly and time consuming. That is why alternative data-driven or corpus-based approaches have been introduced.

4.3.2 Corpus-based Approaches

Corpus-based approaches, that do not strictly follow explicit theory of linguistics, use some algorithms to learn, for example the morphological segmentation of a language, from sample data (corpus). The acquired knowledge is then used to perform the morphological analysis task [Kazakov and Manandhar, 2001]. Corpus-based approaches can be further categorized into supervised [van den Bosch, 1997, Wicentowski, 2004] and unsupervised [Goldsmith, 2000, Creutz and Lagus, 2005] approaches according to the kind of corpus used to train the system. Supervised approaches use annotated text corpora whereas unsupervised ones use a raw, unannotated text. Since there are many languages for which a rule-based morphological analyzer or an annotated corpus does not exist, the unsupervised approach is an appealing alternative. It is a particularly interesting option for under-resourced languages like Amharic. Therefore, recent work focused on unsupervised morphology learning.

As indicated by Goldsmith [2000], there are four approaches used in unsupervised morphology learning. The first approach is the one that identifies morpheme boundaries depending on the degree of predictability of the *nth* letter given the first *n-1* letters. This approach was first proposed by Harris [1955] and further developed by Hafer and Weiss [1974]. The second approach is based on the assumption that local information about a string of letters or words is sufficient to identify morpheme boundaries. Thus, this approach tries to identify bigrams and trigrams that have a high likelihood of being morpheme internal. The third approach concentrates on the discovery of patterns that show the relationship between pairs of related words or paradigms [Goldsmith, 2000]. The fourth one is a top down approach that fo-

cuses on globally optimizing a single criterion, namely the criterion of minimum description length, for the corpus [Jurafsky and Martin, 2008].

The idea in minimum description length (MDL) is that first we try to learn the optimal probabilistic model of some data. The model is then used to assign a likelihood and a compressed length to the entire data set. The proposed model itself is also assigned a length. The MDL principle states that the optimal analysis of the data is the one for which the sum of the data length and the model length is the smallest. The MDL approach to morphology induction has been used by many researchers, notably by Goldsmith [2000] and Creutz and Lagus [2005] who independently developed freely available tools Linguistica and Morfessor, respectively. Since we used Morfessor in our experiment, its brief description follows.

4.3.2.1 Morfessor

Morfessor, developed by Creutz and Lagus [2005], is an unsupervised morphology learner and morpheme segmenter. It learns a morphological segmentation of the word forms in the input data. Unlike many other morphology induction programs such as Linguistica [Goldsmith, 2000], Morfessor tries to identify all the morphemes of a given word. That means it produces a full segmentation of word forms. For example, the word *dessertspoonfuls* is segmented by morfessor as *dessert+spoon+ful+s* instead of *dessertspoonful+s*.

Morfessor has undergone four development steps that resulted in four versions [Creutz, 2006]: Morfessor Baseline, Morfessor Baseline-Freq-Length, Morfessor Categories-ML and Morfessor Categories-MAP. Currently, two versions of the Morfessor program, Morfessor Baseline and Morfessor Categories-MAP, are freely available for research purpose. However, when we started our experiments only the Morfessor Baseline system was available and consequently it is the one used in our experiments.

The Morfessor Baseline model is based on the MDL criterion. It learns a lexicon of morphs which is concise and produces a compact representation for the words in the corpus as defined below.

$$\arg\max p(Lexicon/corpus) = \arg\max p(corpus/Lexicon)p(Lexicon) \qquad (4.1)$$

In Morfessor baseline the $p(Lexicon)$ (which is the prior probability of getting M distinct morphs) is estimated on the basis of two properties of morphs: frequency and character sequence probability (length probability). The morphs are simply strings

of letters and do not have substructure, i.e. the lexicon is flat. The model does not assume a uniform probability distribution for the proposed morphs. Instead the relative frequency of the morphs is considered as their probability. The baseline model is prone to three kinds of errors: under-segmentation or incomplete segmentation which implies that some morpheme boundaries are missed, over-segmentation where words are split into too many parts, and morphotactic violation which occurs when a substring that can function as a morph in some context is proposed in the wrong context. In Morfessor Baseline, a frequent string is most concisely coded in one piece, regardless of its linguistic structure. This sometimes leads to undersegmentation. In contrast, a rare string is best coded in short substrings which causes oversegmentation. Since the model does not assign any grammatical categories to the proposed morphemes and it does not indicate the context in which the morph can occur, it is prone to the third type of error (morphotactic violation). Nevertheless, this model works well and the developers used it in all of their speech recognition experiments. Morfessor baseline produces a better morph segmentation, from a morphological point of view, when provided with a word type list [Creutz, 2006] instead of word token list.

The Morfessor Baseline-Freq-Length model is an extension of the Morfessor Baseline model. It applies Bayesian prior probabilities to the frequency and length distributions of the morphs. The purpose of the frequency prior (which is derived from Zipf's law) is to favour solutions where the frequency distribution of the proposed morphs is in accordance with Zipf's law. The morph length distribution describes the proportion of morphs of a particular length, measured in letters. The Baseline-Freq-Length model utilizes a gamma distribution as a prior for morph length. This model outperforms Morfessor Baseline due to the priors for morph length and frequency. Although the number of over- and under-segmentations is reduced, the model is insufficient for preventing morphotactic violations. Moreover, the difference between Morfessor Baseline and Baseline-Freq-Length diminishes with larger amounts of data [Creutz, 2006]. It has also been indicated that the length prior is more effective than the frequency prior. As a result, in most of their experiments, the developers omitted the frequency prior and consequently, come up with a Baseline-Length model.

Morfessor Categories-ML is a version that tries to reduce errors of the Baseline models caused by their context insensitivity. It introduces a simple morphotactics in order to reduce morphotactic violation errors. In this model, a segmentation produced by one of the Baseline algorithms is reanalyzed using maximum likelihood (ML) optimization and some heuristics. Each morph in the segmented corpus is tagged with one of the following categories: prefix, stem, or suffix based on a few

usage-based features of the morph. In cases where none of the three categories is likely, morphs are tagged with a category "noise". These noise morphs are short segments, which are not morphs at all or not morphemes in the current context. Thus, the presence of noise morphs typically indicates that a word has been over-segmented or that it contains morphotactic violations. Over-segmentation is reduced by applying a heuristic that joins together noise morphs with their neighbors whereas under-segmentation is alleviated by forcing splits of redundant morphs (morphs that contain other morphs found in the lexicon). Thus, in Morfessor Categories-ML, the size of the lexicon is controlled through these heuristics instead of an overall probability function. Once the lexicon has been modified, maximum likelihood re-estimation is applied in order to re-segment and re-tag the corpus. A first order Hidden Markov Model (HMM) has been used for assigning probabilities to each possible segmentation and tagging of a word form. The HMM is intended to model morphotactics that is expressed by means of a regular expression as $word = ((prefix) * stem\ (suffix)*)+$. Since restrictions (such as a suffix may not start a word, a prefix may not end it, a suffix should not occur immediately after a prefix) are used, some morphotactic violation errors that are observed in the Morfessor Baseline models are removed in the Morfessor Categories-ML model [Creutz, 2006].

Morfessor Categories-MAP works similar to Morfessor Categories-ML but has a more sophisticated formulation than Morfessor Categories-ML. As indicated by Creutz [2006], Morfessor Categories-MAP operates on data sets consisting of word tokens whereas Categories-ML works on word types. Moreover, Morfessor Categories-MAP is a complete maximum a posteriori model. That means, unlike the Morfessor Categories-ML, it does not rely on heuristics to determine the optimal size of the lexicon. Instead it utilizes a hierarchical lexicon structure which provides a different mechanism to control over- and under-segmentations. In this model, under-segmentation can be avoided by expanding a lexical item into the sub-morphs it consists of whereas over-segmentation is avoided by expanding substructures as long as they do not contain noise morphs. Morphotactics is handled in a similar fashion as for the Morfessor Categories-ML model.

Morfessor has been evaluated in two manners: directly by comparing to a linguistic gold standard (called direct evaluation hereafter) and indirectly through a speech recognition experiment (called indirect evaluation afterwards) [Creutz, 2006]. In direct evaluation, the proposed placements of morpheme boundaries have been compared to a linguistic gold standard segmentation. The results of the comparison is then presented using precision, recall and F-measure. Precision is the proportion of correct morph boundaries among all morph boundaries suggested by Morfessor

while recall is the ratio of correct boundaries discovered by the algorithm to all morpheme boundaries in the gold standard. F-measure is defined as the harmonic mean of precision and recall as follows[Creutz, 2006].

$$F\text{-}measure = \frac{1}{[\frac{1}{2}(\frac{1}{precision} + \frac{1}{recall})]} \tag{4.2}$$

The Helsinki University of Technology Morphology Evaluation Gold Standard (Hutmegs) [Creutz and Lindén, 2004], consisting of a linguistic segmentation for English and Finnish, has been used in the direct evaluation. The Finnish gold standard contains the segmentation for 1.4 million word types while the English one contains segmentations for 120,000 distinct words. The gold standards include word types, however if a word has several possible segmentations, all of them are supplied.

The evaluation data sets used for Finnish is prose and news text from the Finnish Information Technology center for Science and the Finnish National News Agency. The English data set consists of the Brown corpus, a sample of the Gigaword corpus, as well as prose, news and scientific text from the Gutenberg project. Evaluations have been carried out on data sets containing 10,000, 50,000, 250,000 and 16 million words for Finnish. The same data set sizes are used for English, except for the largest data set which consists of only 12 million words. In the direct evaluation method, Linguistica has also been included (for comparison) besides the four versions of Morfessor (Baseline, Baseline-Length, Categories-ML and Categories-MAP).

For Finnish, it has been found that Categories-ML and Categories-MAP are the best performing algorithms according to the F-measure values. Although Categories-ML is the leader in most cases, for some data set sizes (10,000 and 250,000) the difference between them is not statistically significant. Baseline-Length is slightly better compared to Morfessor Baseline, but the difference is statistically significant only on the 50,000 word data set. The worst performing algorithm on the Finnish data set is Linguistica.

For the English data set, most of the algorithms performed better (in terms of F-measure) than on the corresponding Finnish data set which might be due to the simplicity of English morphology [Creutz, 2006]. Linguistica is the second best performing algorithm on the 50,000 and 250,000 test set. This shows that Linguistica is more suited for a language with simple morphology like English.

Morfessor Baseline, Categories-ML and Categories-MAP have also been evaluated on data sets for Turkish and the Egyptian dialect of Arabic consisting of 17 million tokens or 580,000 types and 150,000 token or 17,000 types, respectively. The gold standard segmentations for Turkish is based on a morphological parser devel-

oped at Bogazici University while the Arabic one is based on a lexicon of Egyptian Colloquial Arabic. However, in their experiment the developers of Morfessor, treated the Arabic word stems as correct morphs although the stems should have been analyzed into root and vowel patterns. Their results show that the performance of Morfessor Baseline varies greatly with data size and language, whereas the others seem to perform constantly irrespective of the language. The F-measure obtained for the Morfessor Baseline model on the Arabic data set (the smallest data set compared to the others) is very small, namely 41.7% [Creutz, 2006].

Only the Morfessor Baseline model is evaluated indirectly through a speech recognition experiment. N-gram language models have been developed using different units (syllables, words, statistical morphs obtained using Morfessor Baseline and grammatical morphs) and the language models have been compared in terms of cross-entropy and as integrated components of a large-vocabulary speaker-dependent speech recognition system. Two experiments have been conducted [Siivola et al., 2003, Hirsimäki et al., 2005] that have already been reviewed in Chapter 2 of this thesis. In general, the results of the experiments showed that statistical morphs perform best.

4.4 Computational Morphology for Amharic

Since 2000, several researchers (pioneered by Abiyot [2000]) have attempted to develop morphological systems for Amharic. The attempts differ in the method they employ, the amount of data they used, the type of problem they tried to solve, etc. The following is a review of works conducted on Amharic computational morphology in chronological order.

Abiyot [2000] developed a prototype automatic word parser for Amharic verbs and their derivations, particularly nouns. His system is a rule based one that is designed based on the morphological features of the language. The parser handles morphotactics, non-concatenative morphology and phonological (spelling) changes. Moreover, the system has the ability of indicating the part-of-speech of words. To perform its tasks, the parser uses different knowledge sources: An Amharic verb root lexicon, an affix database and phonological rules. The performance of the parser has been tested on a test set consisting of 200 verbs and 200 nouns. The parser was able to recognize 86% of the verbs and 84% of the nouns correctly. Apart from the number of words in the test set, nothing is said about the number of roots and affixes in the databases used in the prototype system.

Nega and Willett [2002] developed an Amharic iterative context-sensitive stemmer for processing document and query words in information retrieval. The stemmer

uses a list of stop words and affixes, which have been developed based on statistical methods (but with extensive manual interventions) from a corpus consisting of texts from different domains. The stemmer works, simply, as follows. A word to be stemmed is first checked against the stop word list to ensure its processing. If it is not in the list, it is passed to the prefix and then to the suffix removal modules. After all the prefixes and suffixes have been removed, the resulting stem is used to generate the consonantal roots. The stemmer takes account of letter inconsistencies and reduplicative verb forms. The authors tested the stemmer on a test set consisting of 1221 words and obtained morphologically meaningful stems for 95.9% of the words. The stemmer is more prone to over-stemming (2.7%) than to under-stemming (1.4%). Nega and Willett [2003] showed the effectiveness of this stemmer by applying it to an information retrieval task. As described in [Nega and Willett, 2002], the stemmer produces the stem or the root of a word. Thus, unless it is modified to deliver all the morphemes of a word, a direct application of this stemmer in our work is not possible.

Tesfaye [2002] trained a morphological analysis system for Amharic using a freely available morphology learning program, namely Linguistica [Goldsmith, 2000]. Linguistica requires a large corpus ranging from 5,000 to 1,000,000 words. Tesfaye [2002] used a 5,236 words corpus, the smallest recommended corpus size, to learn the morphology of Amharic using Linguistica. As Linguistica can not handle the root-pattern non-concatenative morphology of Amharic, he developed a stem internal morphological parser (called Amharic Stems Morphological Analyzer - ASMA) based on the theory of autosegmental morphology to analyze the stems identified by Linguistica into their constituent root and pattern morphemes. However, he could not succeed to integrate his stem analyzer to that of Linguistica because of time limitations. Moreover, although the stem analyzer had been developed to analyze the output of Linguistica, Tesfaye [2002] was forced to use a separate corpus consisting of 326 stems for the stem morphological analyzer. The reason behind this, as indicated by the author, is that Linguistica could not produce linguistically correct stems. Both systems use the respective corpora as input and produce morphological dictionaries as their output. To test the performance, the output of the systems (500 words and 255 stems from the output of Linguistica and ASMA, respectively) has been examined by two linguists. 94% of the stems and 87% of the words have been parsed successfully by ASMA and Linguistica, respectively. In the case of Linguistica, undersegmented words are considered as correct analysis.

Sisay and Haller [2003] discussed three related issues about Amharic verb morphology in the context of machine translation, namely Amharic verb classification, aspects of lexical entries for lexical transfer and the implementation of a morpho-

logical analyzer. They used the Xerox finite state tools for implementing a morphological analyzer and indicated that most of the morphological phenomena can be handled using finite state machinery. However, derivational processes that involve the simultaneous application of stem interdigitation and reduplication operations can not be accommodated. The scope of this work is limited to the Amharic verb morphology.

Saba and Gibbon [2005] developed a finite state based computational morphology system for Amharic. As the system is based on a finite state transducer, it can be used for analysis as well as generation. Saba and Gibbon [2005] designed the system in such a way that it can cover all morphological processes of the language: concatenative and non-concatenative (root-pattern and partial or full reduplication). Moreover, they tried to make the analyzer complete by covering words from all parts of speech in the language. However, the number of words in the lexicon is not indicated except for the number of regular verb roots which is 1277. The analyzer takes a string of morphemes as an input and gives the underlying morphemes and morphosyntactic categories. Taking 'Čäräsk' as input, for example, produces '[Črs+VERB+Perf]+Subj+2P+Sg+Masc' as output. As indicated by Saba [2007], the transducer for verbs lacks a deeper analysis of the constraints and consequently is characterized by over-generation. Furthermore, the lexicon of nouns and adjectives is incomplete. We have also noticed the fact that the morphological analyzer developed by Saba and Gibbon [2005] exhibit a dearth of lexicon. It has been tested on 207 words and it analyzed less than 50% (75 words) of the words. In addition, the output of the system is not directly useful for our project which needs the morphemes themselves instead of their morphological features. Since the source code of the analyzer is not yet made available, it is not possible to customize it.

Sisay [2005] developed an Amharic word segmentation system using conditional random fields. The segmentation task did not consider all bound morphemes. While bound morphemes such as prepositions, conjunctions, relative markers, auxiliary verbs, negation markers and coordinate conjunctions are considered; others such as definite article, number, gender and case markers are not considered as segments instead treated as part of a word. Sisay [2005] used five annotated news articles consisting of 1000 words to train and test the system. Using a five-fold cross validation method, he could achieve an accuracy of 84%.

Atelach and Asker [2007] developed a rule-based stemmer for Amharic that reduces words to their citation forms. Although the stemmer is a rule based one, it also employs statistical methods for disambiguation. The authors constructed 65 rules based on the entire Amharic morphology. These rules vary from simple affixation rules to allowed combinations of prefixes and suffixes for each word cat-

egory and set of prefixes. The stemmer works as follows. It first creates a list of all possible segmentations by applying the morphological rules. Each segmentation is then verified by matching each candidate stem against the entries of a machine readable dictionary. If exactly one stem matches the dictionary entry, then that segmentation will be presented as the output of the stemmer. Otherwise, if more than one stem matches, the most likely stem will be selected after disambiguating among the candidate stems based on statistical and other properties of the stem, in particular the length of the stem in terms of the number of characters. On the other hand, if no stem matches the dictionary entry, the stemmer will modify the stem and redo the matching. The authors evaluated the performance of the stemmer on two test sets from different domains: news (1503 tokens) and fiction (470 tokens). The overall accuracy of the stemmer was 76.9% and 60.0% on the news and fiction domain, respectively. Since this stemmer gives the complete segmentation of a word as an output instead of the stem, it would be an appropriate tool for our work which requires all the morphemes of a word. Unfortunately, it was not possible to have access to the stemmer.

Recently, Gasser [2010a] developed a freely available morphological anayser and generator, called HornMorpho, for three Ethiopian languages: Amharic, Oromo and Tigrinya. HornMorpho has been developed as part of the L^3 project at Indiana University which is dedicated to developing computational tools for under-resourced languages. It analyses and generates Amharic verbs as well as nouns, Oromo and Tigrinya verbs. As indicated by Gasser [2010b], given an Amharic word, HornMorpho, returns the root (for verbs only), the lemma and a grammatical analysis in the form of a feature structure description for each possible anaylsis. As it is clear, the analyser became available after we finished our experiments. Moreover, with its current output format, the analyser can not be used directly for our purpose.

As it can be observed from the above review, some of the morphological analysis systems are not directly applicable to our work because of the format of their output and/or their limited coverage. Some others suffer from lack of training data and, therefore, are not efficient for processing large amounts of text as required for language modeling. Those systems that could be used are not available. Therefore, we were forced to find alternative ways for morphological analysis, and decided to use an unsupervised morphological learning system that requires only an unannotated text corpus.

Chapter 5

Morphology-based Language Modeling for Amharic

5.1 Introduction

This chapter expounds the various morphology-based language modeling experiments that we have conducted. The SRI language modeling toolkit has been used to develop the language models. Section 5.2 gives a brief description of the toolkit. As there is no morphological analyzer suited for our purpose and as the development of such a morphological analyzer was not feasible within the time available for the project, we used a freely available unsupervised morphology learning tool, namely Morfessor, to morphologically segment words in our corpus. Section 5.3 presents the morph-based language modeling experiment conducted using the morphs, called statistical morphs hereafter, obtained using the unsupervised morphology induction algorithm. In order to see the benefit of using morphs as a modeling unit, several word-based language models have also been developed and compared with the statistical morph-based language models.

Morfessor deals with concatenative morphology. However, Amharic exhibits the non-concatenative morphological feature besides the concatenative one. Thus, to see the impact of non-concatenative processes in language modeling, we manually segmented a collection of 72,428 word types extracted from a text corpus consisting of 21,338 sentences. Since the morphs have been obtained by applying linguistic rules we call them linguistic morphs. Linguistic morph-based language models have been developed with these data. The performance of such models has been compared with statistical morph-based and word-based language models. The details of the experiment are presented in Section 5.4.

In statistical and linguistic morph-based language modeling, we considered each morph as a unit in language modeling. This, however, leads to a loss of word level dependencies because the span of the n-gram might be limited to a single word. As a solution to this problem and since it allows to integrate any relevant information (e.g. Part-of-Speech tags) to language models, we also developed factored language

models. Section 5.5 describes the development of factored language models and Part-of-Speech taggers that provide additional information for them.

5.2 Modeling Tool

The tool used for language modeling is the SRI Language Modeling toolkit (SRILM) which has been under development in the SRI Speech Technology and Research Laboratory since 1995 [Stolcke, 2002]. SRILM is an extensible toolkit, that runs on UNIX and Windows platforms, for building and applying statistical language models for use in speech recognition, tagging, segmentation, and machine translation.

Besides the standard word-based n-gram language models, it supports the development of other language model types such as class-based models, cache models, factored language models, skipping models, dynamically interpolated models, etc. Moreover, it implements most of the known smoothing techniques. Although SRILM has been originally developed for language model construction and evaluation, over the years it has evolved to include tools that go beyond language model estimation and evaluation, for instance, a tool to rescore and expand lattices.

SRILM is a freely available open source language modeling toolkit. It is currently used by many researchers all over the world and is required by some statistical language processing tools such as Moses.[1]

5.3 Statistical Morph-based Language Models

5.3.1 Corpus Preparation

A text corpus (called ATC_48k hereafter) consisting of 48,090 sentences and 1,542,697 tokens has been prepared. The electronic text has been obtained from ethiozena archive which contains written newscast. Since the target application domain is speech recognition and since we wanted to merge the text corpus with the one prepared by Solomon et al. [2005] for speech recognition, the text has been normalized accordingly. The normalization tasks that we have performed (using manual as well as automatic means) include:

- Correction of spelling and grammar errors. The absence of an Amharic spelling and grammar checker makes the task difficult and laborious. Thus, we do not claim that we corrected every spelling and grammar error in the corpus.

[1] Moses is a statistical machine translation tool.

- Expansion of abbreviations and contractions. The most common contractions are Ethiopian personal names, title and names formed through compounding. For example it is common to write the name ገብረማርያም 'gäbrämaryam' as ገ/ማርያም 'gä/maryam' and the title ጠቅላይ ሚኒስቴር 't'äk'lay minister' which means prime minister as ጠ/ሚኒስቴር 't'ä/minister' or even as ጠ/ሚ 't'ä/mi'. Such contractions have been expanded.

- Separation of concatenated words.

- Textual transcription of numbers. Telephone numbers, date, time, etc. written in numbers have been transcribed.

- Removal of foreign words, particularly English words provided that their removal does not affect the grammatical structure of a sentence.

- Removal of punctuation marks.

After normalization, the ATC_48k corpus has been merged with the text database prepared by Solomon et al. [2005] from the same domain. The combined text corpus (called ATC_120k afterwards) used in the experiment consists of 120,262 sentences or 2,348,150 tokens or 211,120 types. Table 5.1 presents the frequency distribution of words in the ATC_120k corpus.

Frequency	Number of words
1	121285
2 - 10	69526
11 - 100	17356
101 - 1000	2655
1001 - 10000	293
10001 - 20000	3
above 20000	2

Table 5.1: Word frequency distribution

Zipf's law is useful to describe the frequency of words in natural language. It indicates the fact that there are a few very common words, a middling number of medium frequency words, and many low frequency words [Manning and Schütze, 1999]. Our corpus exhibits such a distribution and this implies that the data is sparse. As it can be noted from Table 5.1, more than 50% (121,285) of the words occur only once (hapax legomena) in the corpus. However, our corpus is not the only one to include large number of hapaxes. Zemánek [2001] indicated that CLARA (Corpus Linguae Arabicae), an Arabic corpus, consists of more than 50% hapax

legomena. On the other hand, in our corpus only 5 words appear with a frequency of above 10,000. These words are function words such as ውስጥ /wIsT/ 'in'.

We also compared the frequency distribution of Amharic words with other languages, namely English and German. The negra corpus that consists of 20,602 sentences have been used for German. To make a fair comparison, the same amount of sentences have been taken for English and Amharic from the WSJ and ATC_120k corpora, respectively. Figure 5.1 depicts the frequency distribution of words for the three languages. As can be clearly seen the number of hapax legomena for Amharic is much bigger than for German and English.

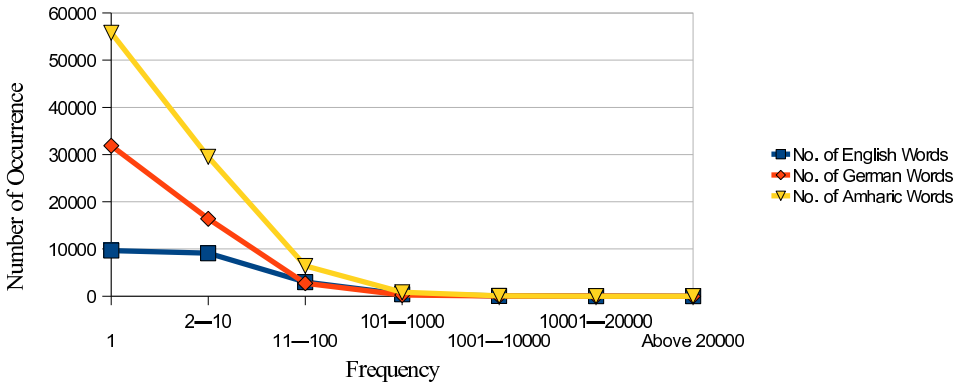

Figure 5.1: Word frequency distribution of Amharic, English and German

5.3.2 Word Segmentation

Developing a sub-word language model requires to have a word parser which splits word forms into its constituent morphs. Different people [Abiyot, 2000, Nega and Willett, 2002, Tesfaye, 2002, Sisay and Haller, 2003, Saba and Gibbon, 2005, Sisay, 2005, Atelach and Asker, 2007] have attempted to develop morphological analyzer for Amharic using different methods as it has been discussed in Chapter 4. However, most of the systems can not be directly used for our purpose. The ones that could be used are not accessible. Hence, there is a need to look for other ways of handling the problem.

An alternative approach is the use of unsupervised corpus-based methods that require only raw unannotated data for morphology induction. The lack of resources such as morphologically annotated corpora for Amharic makes the unsupervised morphology induction methods both interesting and practical. Therefore, two freely available, language independent unsupervised morphology learning tools have been identified: Linguistica [Goldsmith, 2000] and Morfessor [Creutz and Lagus, 2005].

Both tools have been tried on a subset of our corpus (9,996 sentences). Unfortunately, it has been found that Linguistica divides every word into two constituents even if a word actually consists of more than two morphemes. Thus, Morfessor which tries to identify all the morphemes found in a word has been used to produce our morphologically segmented corpus.

Morfessor requires a list of words as an input to learn the morphs. The developers of Morfessor found out that Morfessor, evaluated on Finnish and English data sets, gives better morph segmentation when it is provided with a list of word types instead of a list of word tokens. To compare these findings with the situation in Amharic, two word lists have been prepared from the ATC_120 corpus: a list of tokens and a list of word types. These word lists have then been used as an input to Morfessor. Morfessor learns the morph segmentation from the input data and presents as output the frequency of a word and its constituent morphs (type-based and token-based segmentations). The morphs of a given word are separated by a plus (+) sign. Table 5.2 gives example output from Morfessor.

Input: word type list		Input: word token list	
Frequency	Segmentation	Frequency	Segmentation
1	CAmA[a]	63	CAmA
1	CA + mAcawe	1	CAmA + cawe
1	CAmA + cawene	2	CAmA + cawene
1	CAmA + cawene + nA	1	CAmA + cawene + nA
1	CAmA + cenene	4	CAmA + cenene
1	CAmA + nA	4	CAmA + nA
1	CAmA + ne	3	CAmA + ne
1	CAmA + we	1	CAmA + we
1	CAmA + wene	3	CAmA + wene
1	CAmA + wocacawene	1	CAmA + wocacawene
1	CAmA + woce	4	CAmAwoce
1	CAmA + wocene	2	CAmAwoce + ne
1	CAmA + wocene + nA	1	CAmAwoce + ne + nA
1	CAmA + wocu	1	CAmA + wocu

[a]The transcription is the one employed in Solomon [2006].

Table 5.2: Morfessor segmented data

5.3.2.1 Evaluation of the Word Segmentation

A segmentation performance can be evaluated by comparing morpheme boundaries proposed by the algorithm with a linguistic gold standard. Creutz [2006] indicated that such an evaluation is straightforward and intuitive provided that an adequate

gold standard exists. Since, to our knowledge, there is no gold standard segmentation available for Amharic, we manually prepared one for about 1,000 words. The words have been taken from the output segmentation (type-based segmentation) systematically. That means after taking the first word, every fiftieth word has been taken and the correct segmentation for it has been provided. However, similar to Creutz [2006], we considered Amharic stems as a correct segmentation although they should have been analyzed into root and pattern. The same words have also been selected from the token-based segmentation. The segmentation of words (type-based and token-based segmentation) have been compared with the gold standard and precision, recall and f-measure (see Table 5.3) have been calculated using the evaluation scripts of Hutmegs package [Creutz and Lindén, 2004]. As can be seen from the table, the type-based segmentation is better than the token-based one when compared in terms of f-measure. Moreover, the recall of the type-based segmentation is also higher than the token-based one. However, there is no notable difference in precision. The f-measures for Amharic are higher than the one reported by Creutz [2006] for Arabic (41.7%) using the same version of Morfessor but on a data set consisting of 17,000 distinct words. This might be due to the amount of data used for evaluation and the way the gold standards have been prepared. The Arabic data set is very big compared to the one used to evaluate the Amharic segmentation (only 1,000 word types). While the Arabic gold standard has been generated automatically based on a lexicon of Egyptian Colloquial Arabic, the Amharic one was prepared manually which would result in a more accurate segmentation compared to the automatic method.

Measures	Type-based seg.	Token-based seg.
Precision	82.19%	82.84%
Recall	54.59%	37.60%
F-Measure	65.60%	51.72%

Table 5.3: Evaluation results of the segmentation

5.3.2.2 The Morph Segmented Corpora

The output of Morfessor has been processed (the frequency and the + sign have been removed) and the words in our corpus have been replaced by their constituent morphs automatically. Since Morfessor has been trained on two different word lists, there are two different kinds of output (morph segmentation) and, therefore, two morph-segmented corpora: token_based_corpus and type_based_corpus. Token_based_corpus (consisting of 2,741,218 tokens or 52,337 distinct morphs) is a

morphologically segmented corpus where the morphs have been found by analyzing the list of tokens whereas in type_based_corpus, that consists of 4,035,656 tokens or 15,923 types, the morphs have been obtained by analyzing the word type list. Figure 5.2 shows the frequency distribution of morphs in these corpora. In the token_based_corpus, the majority of morphs (14,550) are in the frequency range of 2 to 5 while in type_based_corpus there are only 2,268 morphs in this frequency range. Most (3,511) of the morphs in type-based corpus have frequencies between 20 and 49.

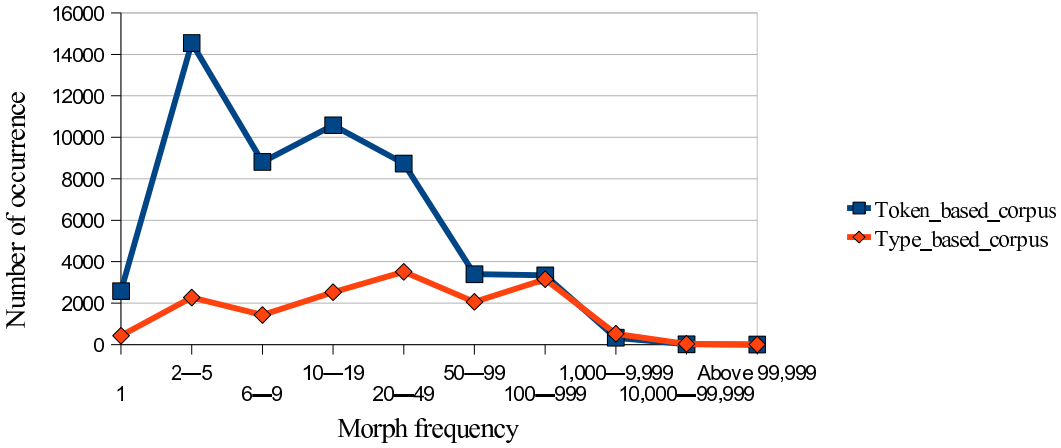

Figure 5.2: Morph frequency distribution

Figure 5.3 depicts the morph length (in terms of number of characters) distribution in token_based_ and type_based_corpus. As can be seen from the figure, the length of most of the morphemes (4,647) in type_based_corpus is six while in token_based_corpus, the majority of morphemes (23,266) consists of 10 to 19 characters. The figure also shows that most of the morphs have even length which is due to the transcription system used. In our corpus each Amharic character is transcribed by a consonant and vowel which made almost all words in the corpus to have even length.

5.3.3 The Language Models

Each morph segmented corpus (token_based_corpus and type_based_corpus described in Section 5.3.2.2) is divided into three parts: training set, development and evaluation test sets with a proportion of 80:10:10. Trigram models with Good-Turing smoothing and Katz-backoff have been developed using the corpora. Although direct comparison of perplexity figures for these models is not possible, a surprisingly big difference in perplexity (860.47 for the token_based_corpus and 117.43 for the

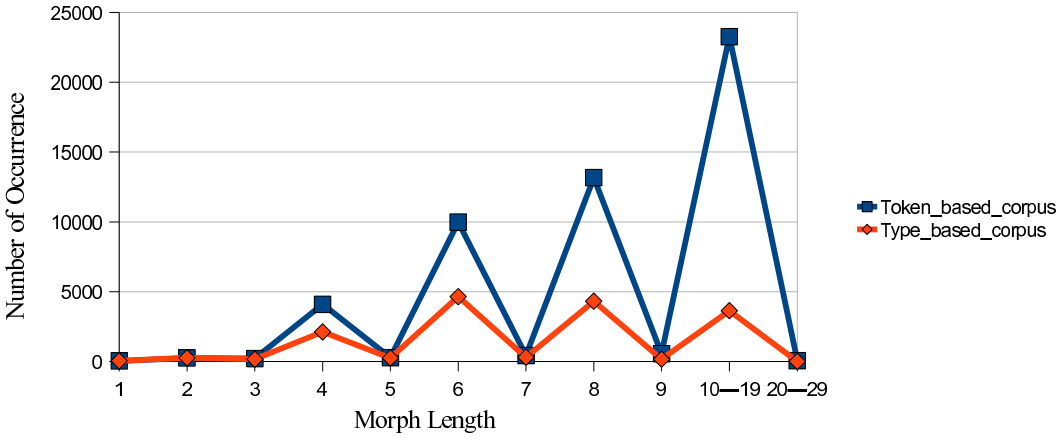

Figure 5.3: Morph length distribution

type_based_corpus) has been observed. The reason for this difference might be the fact that the number of unsegmented words in token_based_corpus (45,767) is greater than that of the type_based_corpus (11,622). This conforms to the finding of Creutz and Lagus [2005] that segmentation is less frequent when a list of word tokens is used as input to Morfessor. For instance, one can observe from Table 5.2 that the word /CAmAwoce/ has been considered as a single morph word albeit it should be segmented into /CAmA/ and /woce/. Moreover, the fact that there are many morphs with length 10 to 19 (see Figure 5.3) further proves that segmentation is less common when word-token list is used as input. From Figure 5.4 we can also see that the vocabulary size has been reduced when word type list has been used as input to morfessor. This is because Morfessor Baseline most concisely codes frequent strings into one piece (regardless of their linguistic structure) which in turn engenders under-segmentation while rare strings are coded in short substrings [Creutz, 2006]. The segmentation evaluation result presented in Section 5.3.2.1 also shows that type-based segmentation is better than the token-based one. Accordingly, only the type_based_corpus has been used for subsequent experimentation and the following results presented in this section are based on this corpus.

N-gram models of order 2 to 5 have been trained using the type_based_corpus. The effect of different smoothing techniques (Good-Turing, Absolute discounting, Witten-Bell, Natural discounting, modified and unmodified Kneser-Ney) on the quality of language models has been studied. The best results obtained for each smoothing technique are presented in Table 5.4.

As it can be seen from Table 5.4, the best performing model is a pentagram model with unmodified Kneser-Ney smoothing. This result is in line with the finding

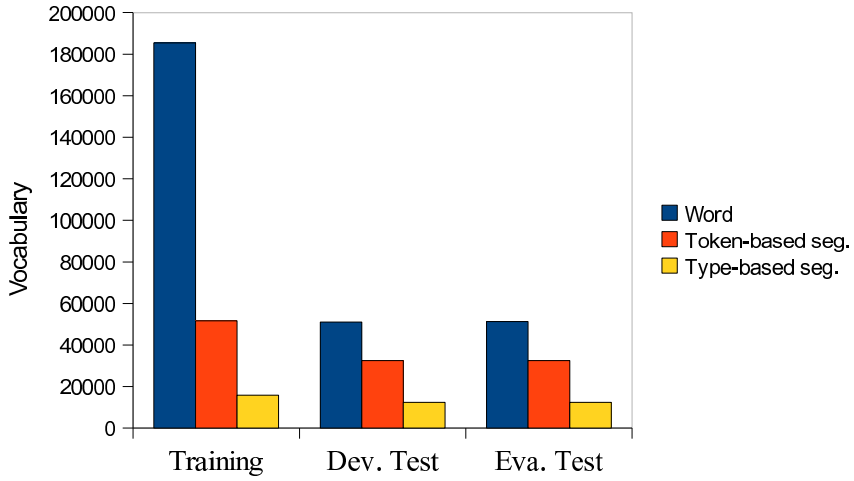

Figure 5.4: Vocabulary size before and after segmentation

N-gram	Smoothing technique	Perplexity
Quadrogram	Good-Turing with Katz backoff	113.24
Pentagram	Absolute Discounting with 0.7 discounting factor	112.79
Pentagram	Witten-Bell	110.88
Pentagram	Natural Discounting	117.37
Quadrogram	Modified Kneser-Ney	107.54
Pentagram	Unmodified Kneser-Ney	103.63

Table 5.4: Perplexity of statistical morph-based language models

of Chen and Goodman [1998] that Kneser-Ney and its variation outperform other smoothing techniques.

Probability estimates of different n-gram order have been interpolated for Witten-Bell, Absolute discounting and modified Kneser-Ney smoothing techniques. Interpolation has been performed only for these three smoothing techniques because the SRILM toolkit supports interpolation only for them. Table 5.5 shows the best results for each smoothing technique.

N-gram	Smoothing technique	Perplexity
Quadrogram	Witten-Bell	112.1
Pentagram	Modified Kneser-Ney	101.38
Quadrogram	Absolute Discounting with 0.7 discounting factor	118.38

Table 5.5: Perplexity results with interpolation

Interpolating n-gram probability estimates at the specified order n with lower order

estimates sometimes yield better models [Stolcke, 2002]. Our experiment verified this fact. A pentagram model with Kneser-Ney smoothing and interpolation of n-gram probability estimates has a perplexity of 101.38. For the other smoothing techniques an increase in perplexity has been observed. This best performing model has a perplexity of 102.59 on the evaluation test set.

As indicated by Stolcke [2002], discarding unknown words or treating them as a special "unknown word" token affects the quality of language models. Thus, unknown words have been mapped to a special "unknown word" token for the best model indicated in Table 5.5 and an increase in perplexity (to 102.26) has been observed. This might be due to the fact that there are only 76 out-of-vocabulary words.

5.3.4 Word-Based Language Model

To compare the results obtained for statistical morph-based language models, we have also developed word-based language models. For this purpose, we used the corpus from which the morph-segmented corpus has been prepared. Table 5.6 shows the perplexity of word-based models. The pentagram model with unmodified Kneser-Ney is the best model compared to the other word-based language models.

N-gram	Smoothing technique	Perplexity
Trigram	Good-Turing with Katz backoff	1151.29
Pentagram	Absolute Discounting with 0.7 discounting factor	1147.04
Pentagram	Witten-Bell	1236
Pentagram	Natural Discounting	1204.14
Quadrogram	Modified Kneser-Ney	1107.32
Pentagram	Unmodified Kneser-Ney	1078.16

Table 5.6: Perplexity of word-based models

Interpolation of n-gram probability estimates has also been tried for the three smoothing techniques for which SRILM supports interpolation. As it can be seen from Table 5.7, improvement has been achieved for a pentagram model with modified Kneser-Ney as a result of interpolation. In contrast, models using the other two smoothing techniques (Witten-Bell and Absolute Discounting) did not benefit from interpolation.

The optimal quality has been obtained with pentagram language model with modified Kneser-Ney, interpolation of n-gram probability estimates, and a mapping of unknown words to a special "unknown word" token. This model has a perplexity of 879.25 and 873.01 on the development and evaluation test sets, respectively.

N-gram	Smoothing technique	Perplexity
Pentagram	Witten-Bell	1241.41
Pentagram	Modified Kneser-Ney	1059.38
Pentagram	Absolute Discounting with 0.7 discounting factor	1158.63

Table 5.7: Perplexity of word-based models with interpolation

Although it is not possible to compare the perplexities of our word-based language models and the ones reported by Solomon [2006] (where the maximum perplexity of a bi-gram word-based language model was 167.89) because of the different test sets used, we did not expect such a big gap in the figures. To discover the reason behind the divergence in perplexity, we have developed word-based language models using our corpus (consisting of 96,205 sentences or 185,468 word types for training and 12,026 sentences or 51,056 distinct words for testing) in the same fashion as Solomon [2006] did. In Solomon [2006] HLStats, HBuild and HSGen modules of the HTK toolkit [Young et al., 2006] have been used. HLStats creates a bigram probability, HBuild converts the bigram language model into lattice format and HSGen generates sentences from the lattice and calculates the perplexity.

Using this method it has been possible to develop a bi-gram word-based language model with a perplexity of 239.45. However, this perplexity figure is still large compared to 167.89 reported by Solomon [2006]. This is due to the nature of the data used in the experiments. In our corpus, there are many long sentences (35,626 and 4,434 sentences in the training and test sets, respectively) that contain more than 20 words. On the other hand, the training data used by Solomon [2006] contains only 51 long sentences (out of 73,895 sentences) while the test set has none. We conducted another experiment following the same procedure but excluding the long sentences from our training and test sets. In this experiment, the training set consists of 60,597 sentences while the test set includes 7,592 sentences or 27,550 word types. Using this data it was possible to get a perplexity of 181.63.

Obviously, the problem with the method used by [Solomon, 2006] is that it calculates the perplexity from automatically generated sentences and there is no guarantee for the correctness of these sentences. In addition, when the test is conducted repeatedly, the perplexity values vary as the sentences generated are different. For example, in the experiment that we conducted with short sentences (containing 20 or less words), the perplexity values varied from 181.63 to 201.27 as shown in figure 5.5. Therefore, we can not directly compare the perplexity of the word-based language model that has been developed in the same way as Solomon [2006] with the one reported in [Solomon, 2006] let alone with the perplexities of the models that

have been tested on real test sentences.

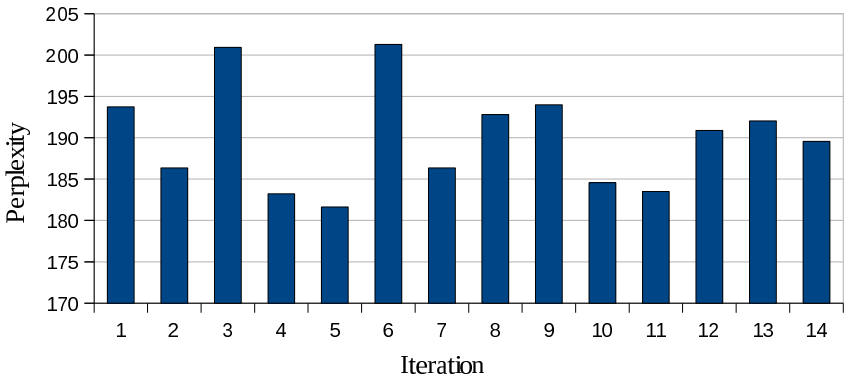

Figure 5.5: Perplexity variance

5.3.4.1 Influence of Data Quality

Although we expect that the high perplexity of our word-based language models is attributed to the morphological richness of the language, spelling errors might also contribute. To estimate the influence of spelling errors, we have conducted two experiments. For these experiments, two data sets have been prepared: data_set_I and data_set_II. 10,000 sentences (consisting of 312,003 word tokens or 60,292 distinct words) of the ATC_48k have been manually checked for spelling errors. The text which is free from spelling errors consists of 10,005 sentences or 307,689 word tokes or 57,687 types. The higher number of sentences in the spelling error free text is due to splitting of two or more sentences which were considered originally a single one. As expected, the number of tokens and word types have also been reduced after spelling error correction. This might roughly show the spelling error rate in the original text. The spelling error free sentences have been merged with the sentences that Solomon et al. [2005] used to develop a speech corpus. We took only 11,917 sentences (that contain 117,792 tokens or 32,382 types) which are actually read for the speech corpus as this will assure us that there is no spelling error. This forms data_set_I that consists of 21,922 sentences and 425,481 tokens. Data_set_II is prepared in the same way except that the spelling errors in the 10,000 sentences have not been corrected. It consists of 21,917 sentences and 429,795 tokens. These data have been divided into training set, development and evaluation test set with a proportion of 80:10:10 and word-based language models have been developed. The perplexity of the models developed using these data sets are presented in Table 5.8 and 5.9.

N-gram	Smoothing technique	Perplexity
Quadrogram	Absolute Discounting with 0.7 discounting factor	981.464
Quadrogram	Witten-Bell	1091.03
Quadrogram	Natural Discounting	1013.81
Trigram	Modified Kneser-Ney	970.285
Trigram	Unmodified Kneser-Ney	940.046

Table 5.8: Word-based models with data_set_I

N-gram	Smoothing technique	Perplexity
Quadrogram	Absolute Discounting with 0.7 discounting factor	988.073
Pentagram	Witten-Bell	1096.71
Quadrogram	Natural Discounting	1022.22
Trigram	Modified Kneser-Ney	986.471
Trigram	Unmodified Kneser-Ney	955.999

Table 5.9: Word-based models with data_set_II

As it can be observed from Table 5.8 and 5.9, the best models are the tri-gram models with unmodified Kneser-Ney smoothing for both data sets. The perplexity values are 940.046 and 955.999 for data_set_I and data_set_II, respectively. When n-gram estimates are interpolated, the four-gram models with modified Kneser-Ney smoothing have the lowest perplexity for both data sets, as shown in Table 5.10 and 5.11.

N-gram	Smoothing technique	Perplexity
Quadrogram	Witten-Bell	1084.92
Quadrogram	Modified Kneser-Ney	936.898
Quadrogram	Absolute Discounting with 0.7 discounting factor	979.125

Table 5.10: Interpolated word-based models data_set_I

Mapping the out-of-vocabulary words to a special “unknown word” token reduced the perplexity of the best performing model developed using data_set_I by 349.487 (from 936.898 to 587.411). This model has a perplexity of 613.983 on the evaluation test set. For data_set_II, a perplexity reduction of 372.632 (from 953.953 to 581.321) has been observed as a result of mapping unknown words to the “unknown word” token. The latter model has a perplexity of 578.627 on evaluation test set.

There is still a very high perplexity for the best models developed using data_set_I, which is free of spelling errors. This enables us to conclude that correcting spelling errors did not reduce the high perplexity of word-based models and,

N-gram	Smoothing technique	Perplexity
Quadrogram	Witten-Bell	1092.23
Quadrogram	Modified Kneser-Ney	953.953
Quadrogram	Absolute Discounting with 0.7 discounting factor	987.89

Table 5.11: Interpolated word-based models data_set_II

therefore, the sole source for the high perplexity is the morphological feature of the language.

5.3.5 Comparison of Word-based and Statistical Morph-based Language Models

The perplexity values of word-based and morph-based models are not directly comparable as the test sets used have quite different token counts. In this case, it is better to consider the probability assigned to the test sets by the models. A model that assigns a high probability is considered a better model. To avoid underflow, log probabilities are used for comparison. The total log probability of the best performing statistical morph-based model (a pentagram model with Kneser-Ney smoothing and interpolation of n-gram probability estimates, indicated in Table 5.4) is -834495, whereas the corresponding word-based model has a total log probability of -705218. Table 5.12 depicts the log probabilities of best statistical morph-based model and the corresponding word based model which has a perplexity of 1059.38 (see Table 5.7).

Model	Log Probabilities
Best performing morph-based model	-834495
Corresponding word-based model	-705218

Table 5.12: Log probabilities I

The best performing word-based language model (pentagram model with unmodified Kneser-Ney, interpolation of n-gram probabilities, and mapping of unknown words to “unknown word” token) has a total log probability of -726095, while the total log probability of the corresponding statistical morph-based model is -836215 although its perplexity is 102.26. Table 5.13 shows this fact. This tells us that word-based models have high log probability and, therefore, are better models although their perplexity is higher. On the other hand, sub-word based language models offer the benefit of reducing the out-of-vocabulary words rate from 13,500 to 76. This is a great achievement, as the out-of-vocabulary words problem is severe in mor-

phologically rich languages in general, and Amharic in particular. Thus, a speech recognition experiment is required to investigate which model is really better and improves the performance of the speech recognition system.

Model	Log Probabilities
Best performing word-based model	-726095
Corresponding morph-based model	-836215

Table 5.13: Log probabilities II

5.4 Linguistic Morph-based Language Models

5.4.1 Manual Word Segmentation

Morfessor tries to find the concatenative morphemes in a given word. However, a word in Amharic can be decomposed into root, pattern and one or more affix morphemes. Since Morfessor does not handle the non-concatenative feature, we manually segmented 72,428 word types found in a corpus of 21,338 sentences (419,660 tokens). By manually segmenting the tokens in the corpus we hoped to obtain an optimistic estimation of what an automatic procedure could achieve at best if it would be available.

To do the segmentation, we used two books as manuals: Baye [2000EC] and Bender and Hailu [1978]. Yimam's book describes how morphemes can be combined to form words. The list of roots in Bender and Hailu [1978] helped us to cross-check the roots that we suggest during the segmentation. Unlike the unsupervised morphology induction program, namely Morfessor, we provided full segmentation for each word including the root and the pattern whenever appropriate as shown in Table 5.14. The tags in parentheses such as (prefix), (root), etc. are not actually used when we develop the linguistic morph-based language models. But they are used particularly for the purpose of preparing the factored data that is required for factored language modeling.

We do not claim the segmentation to be comprehensive. Since a word type list has been used, there is only one entry in case of polysemy or homonymy. For example, the word ጥሩ /t'Iru/ might be an adjective which means 'good' or it might be an imperative verb which has a meaning 'call' (for second person plural). Consequently, the word has two different segmentations. Nevertheless, we provided only one segmentation based on the most frequent meaning of the word in our text. In other words we disambiguated based on the frequency of use in the text. The geminated and non-geminated word forms, which might have distinct meanings and

Word	Segmentation
HenedamahonE	Heneda (prefix) ma (prefix) hwn (root) a (pattern) E
HenedamahonewA	Heneda (prefix) ma (prefix) hwn (root) a (pattern) wA
Henedamahonu	Heneda (prefix) ma (prefix) hwn (root) a (pattern) u
HenedamahonuA	Heneda (prefix) ma (prefix) hwn (root) a (pattern) uA
Henedamakaru	Heneda (prefix) mkr (root) aa (pattern) u
HenedamaleHakete	Heneda (prefix) maleHakete
HenedamalasekAcawe	Heneda (prefix) mls (root) aa (pattern) k Acaw
Henedamalasuteme	Heneda (prefix) mls (root) aa (pattern) u t m
Henedamamalase	Heneda (prefix) ma (prefix) mls (root) aa (pattern)
HenedamanaCa	Heneda (prefix) mnCh (root) aa (pattern) a
Henedamaqedase	Heneda (prefix) maqedase
HenedamaraDA	Heneda (prefix) maraDA
HenedamaraTa	Heneda (prefix) mrT (root) aa (pattern) a
HenedamaraTu	Heneda (prefix) mrT (root) aa (pattern) u
Henedamaraqehute	Heneda (prefix) mrq (root) aa (pattern) hu t

Table 5.14: Manually segmented words

consequently different segmentations, have also been treated in the same manner as the polysemous or homonymous ones. Because the transcription system does not indicate the geminated consonant, for instance, አጠገቡ /?at'ägäbu/ can be treated as an adverb which means 'next to him' or as a verb with a meaning 'they made somebody else full or they satisfied (somebody else)' based on the gemination of the consonant g. This word could have, therefore, been segmented in two different ways: [?at'ägäb + u] if it is an adverb or [?a + t'gb + aa + u] if it is a verb.

Another point is related to the plural forms of words borrowed from Geez[2] which are formed by epenthesis and vocalic changes. For example, the plural form of the noun ኮከብ /kokäb/ - 'star' is ከዋክብት /käwakbIt/. There are also some Amharic words whose plural forms are formed through partial reduplication. For instance, the plural form of the adjective ትልቅ /tIlk'/ - 'big' is ትላልቅ /tIlalk'/. It would be easier if we analyse such words as, for example, käwakbIt = kokäb + NPl[3] or tIlalk' = tIlk' + AdjPl[4]. However, we need an analysis that gives the morphemes themselves. Thus, such words have not been segmented since it is difficult to segment them into morphemes. Nevertheless, we believe that these limitations do not significantly affect the quality of our manually segmented corpus.

Once we have the segmented list of words, we automatically substituted each and

[2]Another Semitic language mostly used in the liturgy of Ethiopian and Eritrean Orthodox täwahdo churches.

[3]A short form for plural noun.

[4]A short form for plural adjective.

every word in our corpus (from which the word type list has been derived) with its segmentation so as to have the linguistic morph-segmented corpus, manually_seg corpus. This corpus has 1,141,434 tokens or 11,154 distinct morphs. As can be seen from Table 5.15, the length of most of the morphs in the manually_seg corpus is between two and six which corresponds to the segmentation found by Morfessor trained on word type list where most of morphs have a length of six. The majority of the morphs in manually_seg corpus have a frequency of five or less which resembles the segmentation trained on the word token list (see Figure 5.6).

Length	No. of Occurrence
1	28
2 - 6	6364
7 - 9	2826
10 - 19	1930
20 - 29	6

Table 5.15: Morph length distribution of manually segmented corpus

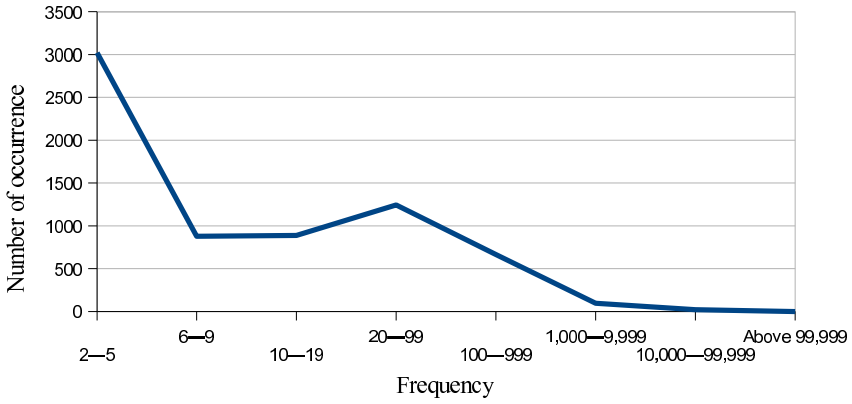

Figure 5.6: Linguistic morpheme frequency distribution

5.4.2 Morph and Word-based Language Models

To find out whether the manual segmentation contributes to an improvement of model quality, we also segmented the same text corpus using Morfessor, and consequently we obtained a statistically morph-segmented version of the corpus, morfessor_seg corpus. The same text corpus (without being morphologically segmented) has also been used to train word-based models. That means we have three versions of the corpus: an unsegmented version for training word based models and a linguistically morph-segmented (manually_seg) as well as a statistically morph-segmented

one (morfessor_seg). The experimental results presented in the next section are based on these three corpora.

Each of the three corpora has been divided into three parts: training set, development and evaluation test sets in the proportion of 80:10:10. Language models with several smoothing techniques and n-gram orders have been trained using the three corpora: manually_seg, morfessor_seg and word_based corpus. The best model for each corpus is presented in Table 5.16.

Corpus	N-gram	Smoothing techniques	Perplexity	Logprob
Manually_seg	6-gram	Unmodified Kneser-Ney	21.597	-153178
Morfessor_seg	5-gram	Unmodified Kneser-Ney	99.49	-161241
Word_based	4-gram	Unmodified Kneser-Ney	1000.86	-116387

Table 5.16: Perplexity difference according to the type of corpus

As it can be clearly seen from Table 5.16, the model that uses the manually segmented data has a lower perplexity[5] compared to all the other models. But, when considering the log probability, the word based model still seems better than the morph-based ones. The models developed with manually segmented data excel the ones developed with automatically segmented data. With regard to out-of-vocabulary words, the models developed using automatically and manually segmented corpus have 45 and 507 out-of-vocabulary words, respectively. In contrast, the number of out-of-vocabulary words in word-based models is very high, namely 4821. One can also see that unmodified Kneser-Ney smoothing outperformed all the other smoothing techniques irrespective of the type of corpus.

Corpus	N-gram	Smoothing techniques	Perplexity	Logprob
Manually_seg	6-gram	Modified Kneser-Ney	20.703	-151071
Morfessor_seg	5-gram	Modified Kneser-Ney	97.297	-160459
Word_based	5-gram	Modified Kneser-Ney	997.387	-116329

Table 5.17: Effect of interpolation

Interpolation brought improvement (see Table 5.17) to all the models regardless of the kind of corpus used. However for the word based model, the increase in log probability is not as large as it is for the others. The language models developed us-

[5]The perplexity and the out-of-vocabulary words reported here for manually segmented data are different from the one reported in Martha and Menzel [2009]. This is because in Martha and Menzel [2009] there were words which were left unsegmented as it was difficult to decide on their proper segmentation. Later these words have all been segmented in consultation with a linguist and according to recommendations in the literature.

ing manually segmented data consistently surpassed (in quality) the ones developed using automatically segmented data.

5.5 Factored Language Models

Sections 5.3 and 5.4 presented the various morph-based language models for Amharic. In these experiments morphemes are considered as units of a language model. This, however, might result in a loss of word level dependencies since the root consonants of the words may stand too far apart. Therefore, approaches that capture word level dependencies are required to model the Amharic language. Kirchhoff et al. [2003] introduced factored language models that can include word level dependencies while using morphemes as units in language modeling. That is why we opted for developing factored language models for Amharic.

As it has been discussed in Chapter 2, in addition to capturing the word level dependencies, factored language models also enable us to integrate any kind of relevant information to a language model. Part of speech (POS) or morphological class information, for instance, might improve the quality of a language model as knowing the POS of a word can tell us what words are likely to occur in its neighborhood [Jurafsky and Martin, 2008]. To see the effect of integrating POS information to the language models, a POS tagger which is able to automatically assign POS information to the word forms in a sentence is needed.

5.5.1 Previous Works on Amharic Part-of-Speech Taggers

Mesfin [2000] attempted to develop a Hidden Markov Model (HMM) based POS tagger for Amharic. He extracted a total of 23 POS tags from a page long text (300 words) which is also used for training and testing the POS tagger. The tagger does not have the capability of guessing the POS tag of unknown words, and consequently all the unknown words are assigned a UNC tag, which stands for unknown category. As the lexicon used is very small and the tagger is not able to deal with unknown words, many of the words from the test set were assigned the UNC tag.

Sisay [2005] developed a POS tagger using Conditional Random Fields. Instead of using the POS tagset developed by Mesfin [2000], Sisay [2005] developed another abstract tagset (consisting of 10 tags) by collapsing some of the categories proposed by Mesfin [2000]. He trained the tagger on a manually annotated text corpus of five Amharic news articles (1000 words) and obtained an accuracy of 74%.

A very recent parallel, but independent development, is due to Gambäck et al. [2009]. There, three tagging strategies have been compared – Hidden Markov Models

(HMM), Support Vector Machines (SVM) and Maximum Entropy (ME) – using the manually annotated corpus [Girma and Mesfin, 2006] developed at the Ethiopian Language Research Center (ELRC) of Addis Ababa University. Since the corpus contains a few errors and tagging inconsistencies, they cleaned the corpus. Cleaning includes tagging non-tagged items, correcting some tagging errors and misspellings, merging collocations tagged with a single tag, and tagging punctuations (such as " and /) consistently. They have used three tagsets: the one used in Sisay [2005], the original tagset developed at ELRC that consists of 30 tags and the 11 basic classes of the ELRC tagset. The average accuracies (after 10-fold cross validation) are 85.56, 88.30, 87.87 for the TnT-, SVM- and maximum entropy based taggers, respectively for the ELRC tagset. They also found that the maximum entropy tagger performs best among the three systems, when allowed to select its own folds. Their result also shows that the SVM-based tagger outperforms the other ones in classifying unknown words and in the overall accuracy for the tagset (ELRC).

As the data sets used to train the first two systems ([Mesfin, 2000] and [Sisay, 2005]) are very small, it is not possible to apply the taggers to large amount of text which is needed for training a language model. Thus, we developed one for our purpose. Section 5.5.2 presents the POS tagger experiment we have conducted for factored language modeling purpose. The third work [Gambäck et al., 2009] is a parallel development, i.e. it has been published after we developed our POS tagger.

5.5.2 Amharic Part-of-Speech Taggers

5.5.2.1 The POS Tagset

In our experiment, we used the POS tagset developed within "The Annotation of Amharic News Documents" project at the ELRC. The purpose of the project was to manually tag each Amharic word in its context [Girma and Mesfin, 2006]. In this project, a new POS tagset for Amharic has been derived. The tagset has 11 basic classes: nouns (N), pronouns (PRON), adjectives (ADJ), adverbs (ADV), verbs (V), prepositions (PREP), conjunction (CONJ), interjection (INT), punctuation (PUNC), numeral (NUM) and UNC which stands for unclassified and used for words which are difficult to place in any of the classes. Some of these basic classes are further subdivided and a total of 30 POS tags have been identified as shown in Table 5.18. Although the tagset contains a tag for nouns with preposition, with conjunction and with both preposition and conjunction, it does not have a separate tag for proper and plural nouns. Therefore, such nouns are assigned the common tag N.

Categories	Tags
Verbal Noun	VN
Noun with prep.	NP
Noun with conj.	NC
Noun with prep. & conj.	NPC
Any other noun	N
Pronoun with prep.	PRONP
Pronoun with conj.	PRONC
Pronoun with prep. & conj.	PRONPC
Any other pronoun	PRON
Auxiliary verb	AUX
Relative verb	VREL
Verb with prep.	VP
Verb with conj.	VC
Verb with prep. & conj.	VPC
Any other verb	V
Adjective with prep.	ADJP
Adjective with conj.	ADJC
Adjective with prep. & conj.	ADJPC
Any other adjective	ADJ
Preposition	PREP
Conjunction	CONJ
Adverbs	ADV
Cardinal number	NUMCR
Ordinal number	NUMOR
Number with prep.	NUMP
Number with conj.	NUMC
Number with prep. & conj.	NUMPC
Interjection	INT
Punctuation	PUNC
Unclassified	UNC

Table 5.18: Amharic POS tagset (extracted from Girma and Mesfin [2006])

5.5.2.2 The Corpus

The corpus used to train and test the taggers is also the one developed in the above mentioned project — "The Annotation of Amharic News Documents" [Girma and Mesfin, 2006]. It consists of 210,000 manually annotated tokens of Amharic news documents.

In this corpus, collocations have been annotated inconsistently. Sometimes a collocation is assigned a single POS tag and sometimes each token in a collocation got a separate POS tag. For example, 'tmhrt bEt', which means *school*, has got a single POS tag, N, in some places and a separate POS tags for each of the tokens in some other places. Therefore, unlike Gambäck et al. [2009] who merged a collocation with a single tag, effort has been exerted to annotate collocations consistently by assigning separate POS tags for the individual words in a collocation.

As the software used for training the taggers requires a corpus that lists a word and its tag (separated by white space) per line, we had to process the corpus accordingly. Moreover, the place where and date on which the news item appeared have been deleted from the corpus as they were not tagged. After doing the above mentioned pre-processing tasks, we ended up with a corpus that consists in 205355 tagged tokens.

5.5.2.3 The Software

We used two kinds of software, namely TnT and SVMTool, to train different taggers. TnT, Trigram'n'Tags, is a Markov model based, efficient, language independent statistical part of speech tagger [Brants, 2000]. It has been applied to many languages including German, English, Slovene, Hungarian and Swedish successfully. Megyesi [2001] showed that TnT is better than maximum entropy, memory- and transformation-based taggers.

SVMTool is a support vector machine based part-of-speech tagger generator [Giménez and Màrquez, 2004]. As indicated by the developers, it is a simple, flexible, effective and efficient tool. It has been successfully applied to English and Spanish.

Since POS tagging is not at the core of this study, we direct readers who are interested to know more about the software, namely TnT and SVMTool, to the respective literature.

5.5.2.4 TnT-Based Taggers

We have developed three TnT-based taggers by taking different amounts of tokens (80%, 90% and 95%) from the corpus as training data and named the taggers as tagger1, tagger2 and tagger3, respectively. Five percent of the corpus (after taking 95% for training) have been reserved as a test set. This test set has also been used to evaluate the SVM-based taggers to make the results comparable.

Table 5.19 shows the accuracy of each tagger. As it is clear from the table, the maximum accuracy was found when 95% of the data (195,087 words) have been used for training. This tagger has an overall accuracy of 82.99%. The results also show that the training has not yet reached the point of saturation and the overall accuracy increases, although slightly, as the amount of training data increases. This conforms with findings for other languages that "... the larger the corpus and the higher the accuracy of the training corpus, the better the performance of the tagger" [Brants, 2000]. One can also observe that improvement in the overall accuracy is affected with the amount of data added. Higher improvement in accuracy has been obtained when we increase the training data by 10% than increasing by only five percent.

Compared to similar experiments done for other languages and the result obtained for Amharic by Gambäck et al. [2009], our taggers have a worse performance. The better results obtained by Gambäck et al. [2009] might be due to better optimization, the use of cleaned data and a 10-fold cross-validation technique to train and evaluate the taggers. Nevertheless, since the development of POS tagger is not the main aim of this study, we still consider the result acceptable for the given purpose.

	Accuracy in %		
Taggers	Known	Unknown	Overall
Tagger1	88.24	48.77	82.70
Tagger2	88.09	48.11	82.94
Tagger3	88.00	47.82	82.99

Table 5.19: Accuracy of TnT taggers

5.5.2.5 SVM-Based Tagger

We trained the SVM-based tagger, SVMM0C0, using 90% of the tagged corpus. To train this model, the default values for the cost parameter C (that controls the trade off between allowing training errors and forcing rigid margins) and for other features like the size of the sliding window have been used. The model has been trained in a one pass, left-to-right and right-to-left combined, greedy tagging scheme. The resulting tagger has an overall accuracy of 84.44% (on the test set used to evaluate the TnT-based taggers) as Table 5.20 shows.

A slight improvement of the overall accuracy and the accuracy of known words has been achieved setting the cost parameter to 0.1 (see SVMM0C01 in Table 5.20). The accuracy improvement for unknown words is bigger (from 73.64 to 75.30) compared to the accuracy of known words and the overall accuracy. However, when the cost parameter was increased above 0.1, the accuracy declined. Neither a cost parameter of 0.3 (SVMM0C03) nor of 0.5 (SVMM0C05) brought an improvement in accuracy both for the overall accuracy and for the accuracy of known and unknown words.

	Accuracy in %		
Taggers	Known	Unknown	Overall
SVMM0C0	86.03	73.64	84.44
SVMM0C01	86.97	75.30	85.47
SVMM0C03	86.71	73.49	85.01
SVMM0C05	86.48	71.97	84.61

Table 5.20: Accuracy of SVM-based taggers

To determine how the amount of training data affects accuracy, we trained another

SVM-based tagger (SVMM0C01-95) using 95% of the data and the cost parameter of 0.1. As it can be seen from Table 5.21, only a slight improvement in the overall accuracy and the accuracy for classifying unknown words has been achieved compared to the SVMM0C01 tagger which has been trained on 90% of the data. This corresponds to the findings for TnT-based taggers that improved only marginally when a small amount of data (5%) is added. For known words the accuracy declined slightly. Although this tagger is better (in terms of the overall accuracy) than all the other ones developed in our experiment, it performs still worse than the one reported by Gambäck et al. [2009]. Another tagger (SVMM0C03-95) has been developed using the same data but with a different cost parameter (0.3). However, no improvement in performance has been observed.

	Accuracy in %		
Taggers	Known	Unknown	Overall
SVMM0C01-95	86.95	75.35	85.50
SVMM0C03-95	86.76	73.40	85.09

Table 5.21: Accuracy of SVM-based taggers trained on 95% of the data

5.5.2.6 Comparison of TnT- and SVM-Based Taggers

The SVMM0C0 has been trained with the same data that has been used to train the TnT-based tagger, tagger2. The same test set has also been used to test the two types of taggers so that we can directly compare the results and decide which algorithm to use for tagging our text for factored language modeling. As it can be seen from Table 5.20, the SVM-based tagger has an overall accuracy of 84.44%, which is better than the result we found for the TnT-based tagger (82.94%). This finding is in line with what has been reported by Giménez and Màrquez [2004]. We also noticed that SVM-based taggers have a better capability of classifying unknown words (73.64%) than a TnT-based tagger (48.11%) which has also been reported by Gambäck et al. [2009].

With regard to speed and memory requirements, TnT-based taggers are more efficient than the SVM-based ones. A SVM-based tagger tags 366.7 tokens per second whereas the TnT-based tagger tags 114083 tokens per second. Moreover, the TnT-based tagger, tagger2, requires less (647.68KB) memory than the SVM-based tagger, SVMM0C0, (169.6MB). However, our concern is on the accuracy of the taggers instead of their speed and memory requirements. Thus, we preferred to use SVM-based taggers to tag our text for the experiment in factored language modeling.

Therefore, we trained a new SVM-based tagger using 100% of the tagged corpus

based on the assumption that the increase in the accuracy (from 85.47 to 85.50%) observed when increasing the training data (from 90% to 95%) will continue if more training data are added. Again, the cost parameter has been set to 0.1 which yielded good performance in the previous experiments. It is this tagger that was used to tag the text for training factored language models.

5.5.3 Amharic Factored Language Models

5.5.3.1 Factored Data Preparation

To train factored language models we need a corpus in which each word is represented as a vector of factors or features. In our experiment each word is considered a bundle of features including the word itself, as well as its part of speech tag, prefix, root, pattern and suffix (cf. Table 5.22).

W-HenedamahonE:POS-VP:PR-Henedama:R-hwn:PA-a:SU-E
W-HenedamahonewA:POS-VP:PR-Henedama:R-hwn:PA-a:SU-wA
W-Henedamahonu:POS-VP:PR-Henedama:R-hwn:PA-a:SU-u
W-HenedamahonuA:POS-VP:PR-Henedama:R-hwn:PA-a:SU-uA
W-Henedamakaru:POS-VP:PR-Heneda:R-mkr:PA-aa:SU-u
W-Henedamakatale:POS-VP:PR-Henedama:R-ktl:PA-aa:SU-null
W-HenedamaleHakete:POS-VP:PR-Heneda:R-null:PA-null:SU-null
W-HenedamalasekAcawe:POS-VP:PR-Heneda:R-mls:PA-aa:SU-kAcaw
W-Henedamalasuteme:POS-VP:PR-Heneda:R-mls:PA-aa:SU-utm
W-Henedamamalase:POS-VP:PR-Henedama:R-mls:PA-aa:SU-null
W-HenedamanaCa:POS-VP:PR-Heneda:R-mnCh:PA-aa:SU-a
W-Henedamaqedase:POS-VP:PR-Heneda:R-null:PA-null:SU-null
W-HenedamaraDA:POS-VP:PR-Heneda:R-null:PA-null:SU-null
W-HenedamaraTa:POS-VP:PR-Heneda:R-mrT:PA-aa:SU-a
W-HenedamaraTu:POS-VP:PR-Heneda:R-mrT:PA-aa:SU-u
W-Henedamaraqehute:POS-VP:PR-Heneda:R-mrq:PA-aa:SU-hut

Table 5.22: Factored representation

Each feature in the feature vector is separated by a colon (:) and consists of a <tag>-<value> pair. In our case the tags are: W for word, POS for Part-of-Speech, PR for prefix, R for root, PA for pattern and SU for suffix. Although, in Amharic words can have more than one prefix and suffix, we considered each word as having zero or one prefix and/or suffix by concatenating a sequence of affixes into a single unit. A given tag-value pair may be missing from the feature bundle. In this case, the tag takes a special value 'null'.

The manually segmented data (described in Section 5.4.1 that include 72,428 word types) has also been used to prepare the factored version of the corpus con-

catenating sequences of prefixes or suffixes into a single unit. This decreased the number of morph tokens to 936,483 and increased the number of distinct morphs to 14,093 (compared to the manually_seg corpus that includes 1,141,434 token or 11,154 types). However, the morph length distribution is similar to the manually_seg corpus (see Table 5.23). Most of the morphemes consists of two to six characters. The curve for the frequency distribution is also similar as shown in Figure 5.7.

Length	No. of occurrence
1	19
2 - 6	7979
7 - 9	3763
10 - 19	2326
20 - 29	6

Table 5.23: Linguistic morpheme length distribution after affix concatenation

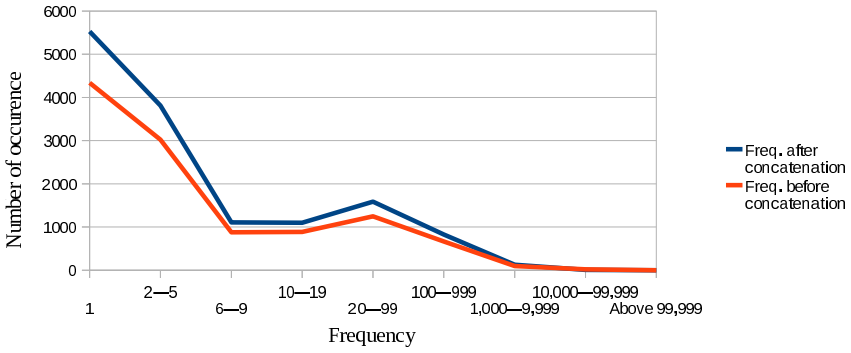

Figure 5.7: Linguistic morpheme frequency distribution before and after affix concatenation

The manually segmented word list has been converted to the format indicated in Table 5.22 and the words in the corpus have been automatically substituted with their factored representation. The resulting corpus has then been used to train and test various kinds of factored language models presented in this section. As we did in other experiments, we divided the corpus into training, development and evaluation test sets in the proportion of 80:10:10. Moreover, since our previous experiments revealed the fact that Kneser-Ney smoothing outperforms all other smoothing methods, we smoothed the factored language models with this technique unless it became impossible to use it due to the occurrence of a zero count of counts.

5.5.3.2 Capturing Word-level Dependency

In Amharic the root consonants of words (derived from consonantal root) represent the basic lexical meaning of the word. Since our morph-based language models (both statistical and linguistic ones) consider each morph as a unit, a loss of word level dependencies could occur as the root consonants of the words stand too far apart. Thus, we have to find a means of capturing word level dependencies while using morphs in language modeling. We have proposed and developed root-based language models in the framework of factored language modeling. One can also consider the models as skipping ones since we skip other morphs during language model estimation.

However, although all the verbs are derived from root consonants, there are words in other part of speech classes which are not derivations of root consonants. Normally, these words have the value 'null' for the root feature (for the R tag introduced in Section 5.5.3.1). When we consider only roots in language modeling, these words will be excluded from our model which in turn will have a negative impact on the language models we develop. Preliminary investigation also revealed the fact that the perplexities of the models have been influenced by the null values for the root tag in the data. Therefore, stems are included to the models as shown in equation 5.1. We did not introduce a new stem feature into our factored data representation, but considered the stem as a root, if the word is not a derivation of consonantal root.

$$r_i \equiv \begin{cases} r_i, & if\ root \neq null \\ stem_i, & otherwise \end{cases} \tag{5.1}$$

where $stem_i$ is the stem of a word (that is not derived from root consonants) after removing all the prefixes and suffixes.

We have trained root-based models of order 2 to 5. Table 5.24 shows the perplexity of these models on the development test set. A higher improvement in perplexity (278.565 to 223.259) has been observed when we move from bigram to trigram. However, as n increases above 3, the level of improvement declined. This might be due to the small set of training data used. As n increases, n-grams might not appear in the training corpus and, therefore, the probabilities are computed on the basis of the lower order n-grams. The pentagram model is the best model compared to the others. This model has a perplexity of 204.946 on the evaluation test set.

In order to determine the benefit gained from the root-based model, we developed word based models with the same training data. They differ from the root-based

Language models	Perplexity
Bigram	278.565
Trigram	223.259
Quadrogram	213.144
Pentagram	211.929

Table 5.24: Perplexity of root-based models on the development set

models in the use of words as a unit instead of roots. These models have also been tested on the same test data (consisting of 2134 sentences or 20989 words) and the same smoothing technique has also been applied. The number of out-of-vocabulary words is much lower (295) in the root-based model than in the word-based one (2672). The pentagram word based model has a perplexity of 972.58 (as shown in Table 5.25) on the development test set. Moreover, the log-probability of the best root-based model is higher (-53102.3) than that of the word based model (-61106.0). However, the perplexities of and the probablities assigned to the test set by the root-based and word based language models are incommensurable since a root might stand for many words and, consequently, the root based models might be less constraining. Therefore, a speech recognition experiment is required to study the benefit of root-based models.

Language models	Perplexity
Bigram	1148.76
Trigram	989.95
Quadrogram	975.41
Pentagram	972.58

Table 5.25: Perplexity of word only models

5.5.3.3 POS Information in Language Modeling

A crucial advantage of factored language models is that they enable us to integrate any kind of relevant information to language models. We exploited this advantage by integrating POS tags into the word based language models. Instead of computing the n-gram probabilities of words only on the basis of previous n-1 words ($w_n|w_{n-2}w_{n-1}$) we calculated the probabilities of words on the basis of n-1 previous words and their POS as ($w_n|w_{n-2}pos_{n-2}w_{n-1}pos_{n-1}$). Since the POS tag of a word can tell us what words are likely to occur in its neighborhood, we expect the models that take POS into account to be precise.

We have developed factored language models with two ($w_n|w_{n-1}pos_{n-1}$) and

four ($w_n|w_{n-2}pos_{n-2}w_{n-1}pos_{n-1}$) parents. In these models a fixed backoff strategy has been applied. The best model (in terms of perplexity) is the model with four parents for which the backoff path has been defined by dropping first pos_{n-2}, then w_{n-2}, then w_{n-1}, and finally pos_{n-1}. This model has a perplexity of 910.37 (see Table 5.26) on the development test set. Since we used the same training and test sets as for the word based models described in Section 5.5.3.2, we can compare perplexity results directly. As it has been already indicated, a pentagram word only model[6] has a perplexity of 972.58 which is higher than the perplexity we have got for the trigram model that considered the previous words' POS tag.

Language models	Perplexity
Word based	972.58
W|W1,POS1	1054.92
W|W2,POS2,W1,POS1	885.81
FLM learned with GA	1119.58

Table 5.26: Perplexity of models with POS

In this experiment we defined the factor combinations to be word and POS and we used a fixed backoff order. However, it is difficult to determine which factor combination and which backoff path would result in a robust model that will bring improvement to a target application. Thus, Duh and Kirchhoff [2004] developed a genetic algorithm (GA) to find out a robust model. We used this algorithm to determine the backoff path that results in a robust model and trained a model with four parents which estimates p($w_n|w_{n-2}pos_{n-2}w_{n-1}pos_{n-1}$). The best model developed with the GA did not consider pos_{n-1} as a history when computing probabilities. This model has a perplexity of 1119.58 which is higher than the one we have got using a fixed backoff strategy and also higher than the perplexity of a word-only model.

5.5.3.4 Other Factored Language Models

Since POS contributed a lot to the improvement of language model quality (in terms of perplexity reduction), n-gram models that take other factors instead of POS into account have been developed. These models are equivalent (in structure) to the best model that include POS in n-gram history during language model estimation. The backoff order is also defined in similar manner. For example, the model that includes the root as an additional factor is estimated as $w_n|w_{n-2}R_{n-2}w_{n-1}R_{n-1}$ and

[6]Note: here we compared a pentagram word only model with bigram and trigram FLM.

the backoff path has been defined by dropping first R_{n-2}, then w_{n-2}, then w_{n-1}, and finally R_{n-1}.

The perplexities of these models are shown in Table 5.27. All the models that take an additional factor in the history outperformed the trigram word only model (see Table 5.25). The model that takes the prefix as an extra information surpassed all the other models including the one with POS. This shows that prefixes are strong in predicting a given word. Pellegrini and Lamel [2006a], who applied word decomposition in Amharic speech recognition, also found a higher word error rate reduction when only the prefixes have been detached from words. The pattern (PA) seems weak in predicting a given word. This finding is in line with that of Kirchhoff et al. [2006] who reported that morphological patterns did not contribute any information for Arabic conversational speech recognition.

Language models	Perplexity
W|W2,PR2,W1,PR1[a]	857.61
W|W2,R2,W1,R1	896.59
W|W2,PA2,W1,PA1	958.31
W|W2,SU2,W1,SU1	898.89

Table 5.27: Perplexity of models with different factors

[a]W = Word, PR = Prefix, R = Root, PA = Pattern, SU = Suffix.

Models that consider all the available features (word, POS, prefix, root, pattern and suffix) as histories have also been developed under the assumption that the improvement (in perplexity) obtained as a result of using an extra feature of a word in the history, will continue if we use more features. A fixed backoff strategy has been applied, in which the factors have been dropped in the following order: suffix, pattern, root, prefix, word and POS. As shown in Table 5.28, a model that considered factors at position n-1 (W1,POS1,PR1,R1,PA1 and SU1) has a perplexity of 1044.41 which is lower than the equivalent model that includes only POS in the n-gram history. Extending the history by adding the factors at n-2 position decreased the perplexity to 886.89. However, as it can be observed from Table 5.27, there are models that take only one extra feature but have a perplexity lower than this model. The reason behind this is not clear at the moment.

As it is difficult to find out which factor combination and which backoff path would result in a robust model, we again used the genetic algorithm [Duh and Kirchhoff, 2004] to learn the best language model structure. The first and second best language models have perplexities of 1075.57 and 1084.29, respectively. These two models use the same factors as histories (POS1 PR1 R1 PA1 SU1 POS2 SU2) but differ in the backoff path employed.

Language models	Perplexity
W|W1,POS1,PR1,R1,PA1,SU1	1044.41
W|W2,POS2,PR2,R2,PA2,SU2,W1,POS1,PR1,R1,PA1,SU1	886.893
First best	1075.57
Second best	1084.29

Table 5.28: Perplexity of models with all factors

The results in Table 5.28 imply that factors are relevant when they are used together with words, cf. the perplexities of the model that considered factors at position n-1 (W1,POS1,PR1,R1,PA1 and SU1) and the first and second best models learnt with the genetic algorithm that did not consider words in the n-gram history.

Generally, our experiment revealed that perplexity improvement can be gained as a result of using extra features of a word in language modeling. However, more investigation is required to ascertain whether this perplexity improvement also results in an improvement in the performance of an application for which the language models are designed. In our case, we need to conduct an experiment to find out whether these language models improve the performance of a speech recognition system or not.

Chapter 6

Application of Morphology-based Language Models

In Chapter 5 we evaluated language models mainly using an intrinsic evaluation metric, namely perplexity because this is the preferred metric for practical language model construction. However, the value of perplexity as a quality measure is not without limitation. Improvement in perplexity does not always lead to an improvement in the performance of an application that uses the language model. Moreover, when we have different token counts in test sets, perplexities are not comparable. In such cases, language models have been compared on the basis of the probability they assign to a test set. Since the best way of evaluating the quality of a language model is to apply it to a specific application for which the language model is designed and see if it results in an improvement in performance, the morphology-based language models have been evaluated by applying them in an Amharic speech recognition system. This chapter presents the results of the speech recognition experiments that we have conducted.

Language models have been applied to a speech recognition system in two different ways. The first is a lattice rescoring framework in which morphemes are used only in the language modeling component, while the lexicon still consists of words. Therefore, this method can not help to analyse the contribution of morpheme-based language models to a reduction of the OOV rate. On the other hand, it avoids the effect of acoustic confusability on the performance of a speech recognition system. Details of the experiment and its results are presented in Section 6.1. The second approach uses morphemes as lexical and language modeling units of the speech recognition system. Although this method suffers from acoustic confusability and limited n-gram language model scope, it enables us to see the contribution of morphemes to the reduction of the OOV rate. Section 6.2 presents the results of this experiment.

6.1 Lattice Rescoring with Morphology-based Language Models

In order to analyse the contribution (in terms of performance improvement) of the morphology-based language models in a speech recognition task, the speech recognition system which has been developed by Solomon [2006] has been used as a baseline in the lattice rescoring experiment. Section 6.1.1 gives a brief description of this baseline speech recognition system. To make our results comparable, we have developed various morphology-based language models (morpheme-based and factored ones) that are equivalent to the ones elucidated in Chapter 5 but have been trained on the text corpus which has also been used to develop the baseline language model. These language models, interpolated with the baseline language model, have then been used in lattice rescoring. We have used interpolated models since previous research (Whittaker and Woodland [2000], Kirchhoff et al. [2002]) also obtained an improvement in WRA using interpolated models. The morpheme-based language models and the result of lattice rescoring using them are presented in Section 6.1.2. Section 6.1.3 exhibits the factored language models and their application for lattice rescoring.

6.1.1 The Baseline Speech Recognition System

6.1.1.1 The Speech and Text Corpus

The speech corpus used to develop the baseline speech recognition system is a read speech corpus [Solomon et al., 2005]. It contains 20 hours of training speech collected from 100 speakers who read a total of 10,850 sentences (28,666 tokens). Compared to other speech corpora that contain hundreds of hours of speech data for training, this corpus is obviously small in size and accordingly the models will suffer from a lack of training data.

The corpus includes four different test sets (5k and 20k both for development and evaluation). However, since our aim is to analyse the contribution of morphology-based language models in speech recognition performance improvement, we have generated the lattices only for the 5k development test set which includes 360 sentences read by 20 speakers.

The text corpus used to train the baseline backoff bigram language model consists of 77,844 sentences (868,929 tokens or 108,523 types).

6.1.1.2 The Acoustic, Lexical and Language Models

The acoustic model is a set of intra-word triphone HMMs with 3 emitting states and 12 Gaussian mixtures that resulted in a total of 33,702 physically saved Gaussian mixtures. The states of these models are tied, using decision-tree based state-clustering that reduced the number of triphone models from 5,092 logical models to 4,099 physical ones.

The Amharic pronunciation dictionary has been encoded by means of a simple procedure that takes advantage of the orthographic representation (a consonant vowel syllable) which is fairly close to the pronunciation in many cases. There are, however, notable differences especially in the area of gemination and insertion of the epenthetic vowel.

The baseline language model is a closed vocabulary (for 5k) backoff bigram model developed using the HTK toolkit. The absolute discounting method has been used to reserve some probabilities for unseen bigrams where the discounting factor, D, has been set to 0.5, which is the default value in the HLStats module. The perplexity of this language model on a test set that consists of 727 sentences (8,337 tokens) is 91.28.

6.1.1.3 Performance of the Baseline System

We generated lattices from the 100 best alternatives for each sentence of the 5k development test set using the HTK tool and decoded the best path transcriptions for each sentence using the lattice processing tool of SRILM [Stolcke, 2002]. Word recognition accuracy of this system was 91.67% with a language model scale of 15.0 and a word insertion penalty of 6.0. The better performance (compared to the one reported by Solomon [2006], 90.94%, using the same models and on the same test set) is due to the tuning of the language model and the word insertion penalty factors.

6.1.2 Morpheme-based Language Models

We have developed several sub-word based language models for Amharic using the morphologically segmented version of the data that has been used to develop the baseline language model.

Both statistical and linguistic morphs have been used as units in language modeling. Morfessor [Creutz and Lagus, 2005] has been applied to produce the statistical morphs. The linguistic morphs have been obtained according to the manually segmented collection of word types described in Section 5.4.1. We substituted each word in the corpus with its segmentation if the word is found in the manually segmented

word collection. Otherwise, the word is left unsegmented. Due to the simplicity of this approach, a substantial share of words (12.3%) in our training data could not be segmented at all.

Using the statistical and linguistic morphs as units, we tried to develop n-gram language models of order two to four. In all cases the SRILM toolkit has been used to train the language models. We smoothed the language models using modified Kneser-Ney smoothing, unless one or more of the required counts of count is/are zero. If one of them becomes zero, then other smoothing techniques that do not depend on such values (e.g. Whitten-Bell) have been used. Table 6.1 presents the perplexity of the various morpheme-based language models on the segmented version of the test set that has been used to test the baseline bigram language model.

Language models	Perplexity	Logprob
Linguistic morph bigram	36.55	-34654.5
Linguistic morph trigram	23.09	-30232.5
Linguistic morph quadrogram	18.39	-28038.5
Statistical morph bigram	114.92	-31800.2
Statistical morph trigram	71.61	-28630
Statistical morph quadrogram	64.22	-27899.7

Table 6.1: Perplexity of morpheme-based language models

As the number of morphs in the linguistically and statistically segmented test sets are different, we can not directly compare the perplexities of statistical and linguistic morpheme-based language models. Although the statistical morpheme-based models have high perplexities, surprisingly they seem better than the linguistic morpheme-based ones if compared with respect to the probability they assign to the test set.

6.1.2.1 Lattice Rescoring with Morpheme-based Language Models

The lattices generated as indicated in Section 6.1.1.3 have been rescored using the various morpheme-based language models and decoded to find the best path. An improvement in word recognition accuracy (WRA) has been observed (see Table 6.2). However, the linguistic morpheme-based models contribute more to the performance improvement (an absolute 0.25% increase in accuracy with the linguistic morph trigram model) than the statistical morpheme-based ones that fared better than the linguistic morpheme-based models with regard to the probability they assign to the test set. Using higher order n-gram brings only a slight improvement in performance, from 91.77 to 91.82 and then to 91.85 as a result of using trigram and

quadrogram language models, respectively.

Language models used	Word recognition accuracy in %
Baseline word-based (BL)	91.67
BL + Statistical morph bigram	91.77
BL + Statistical morph trigram	91.82
BL + Statistical morph quadrogram	91.85
BL + Linguistic morph bigram	91.87
BL + Linguistic morph trigram	91.92
BL + Linguistic morph quadrogram	91.89

Table 6.2: WRA improvement with morpheme-based language models

6.1.3 Factored Language Models

The manually segmented data has also been used to obtain a factored version of the corpus used to develop the baseline language model. The factored version of the corpus has been prepared in a way similar to the one described Section 5.5.3.1. This corpus has then been used to train various kinds of closed vocabulary factored language models[1]. All the factored language models have been tested on the factored version of the test set used to test the baseline language model.

We have developed root-based n-gram language models of order 2 to 5. The perplexity[2] of these models on the development test set is presented in Table 6.3. The highest perplexity improvement has been obtained when the n-gram order has been changed from bigram to trigram.

Language models	Perplexity	Logprob
Root bigram	113.57	-18628.9
Root trigram	24.63	-12611.8
Root quadrogram	11.20	-9510.29
Root pentagram	8.72	-8525.42

Table 6.3: Perplexity of root-based models

We also developed a factored language model that considered all the available factors (word, POS, prefix, root, pattern and suffix) as histories and that uses a fixed backoff

[1]This experiment followed similar procedure with the one reported in [Martha et al., 2009], but the data was modified. Words which are not derived from root consonants have got null value for the R tag in [Martha et al., 2009] while, in the experiment reported here, these words assigned the stem of a word for R (root) feature. Hence, the figures reported are different.

[2]These numbers are not comparable to the perplexities of other models since a root can represent a set of words and consequently root-based models are less constraining.

path by dropping suffix first, then pattern, and so on. Since it is difficult to determine which factor combination and which backoff path would result in a robust model yielding an improvement of speech recognition, we applied the genetic algorithm [Duh and Kirchhoff, 2004] again, to find the optimal one. The best model is the one that uses four factors (word, prefix, root and pattern) as histories and combines generalized all-child and constrained-child backoff. We applied the two best (in terms of perplexity) models, that differ only in the backoff path, to the speech recognition task. The perplexities and log-probabilities of the factored language models are given in Table 6.4. The FLM with fixed backoff is the best model compared to the others.

Language models[a]	Perplexity	Logprob
FLM with fixed backoff	55.04	-15777.7
1st Best factor combination	71.89	-16828.7
2nd Best factor combination	118.62	-18800.3

[a]The language models are smoothed with Witten-Bell smoothing since it was not possible to use the Kneser-Ney smoothing technique due to the existence of zero count of count.

Table 6.4: Perplexity of factored language models

Other factored language models that take one word feature (besides the words) in the n-gram history have been developed. The additional features used are part-of-speech (POS), prefix (PR), root (R), pattern (PA) and suffix (SU). The models developed contain two $(w_n|w_{n-1}X_{n-1})$[3] and four $(w_n|w_{n-2}X_{n-2}w_{n-1}X_{n-1})$ parents where the backoff paths have been defined by dropping w_{n-1} and then X_{n-1} for the former models and by dropping X_{n-2}, then W_{n-2}, then W_{n-1}, and finally X_{n-1} for the latter ones. The perplexity and log-probability of these models are presented in Table 6.5. The models with four parents are almost similar in perplexity and probability. Among the models with two parents, the one that takes root (R) as an additional information is the best one while the model that takes pattern is the worst.

6.1.3.1 Lattice Rescoring with Factored Language Models

Since it is problematic to use factored language models in standard word decoders, we substituted each word in the lattice with its factored representation. A word bigram model that is equivalent to the baseline word bigram language model has been trained on the factored data and used as a baseline for factored representations. This language model has a perplexity of 63.59. The best path transcription

[3]X is a place holder which can represent one of the extra word feature.

Language models	Perplexity	Logprob
W|W1,POS1	64.11	-16377.7
W|W1,PR1	65.02	-16433.2
W|W1,R1	62.14	-16255.1
W|W1,PA1	66.50	-16522.1
W|W1,SU1	65.43	-16458.3
W|W2,POS2,W1,POS1	10.614	-9298.57
W|W2,PR2,W1,PR1	10.67	-9322.02
W|W2,R2,W1,R1	10.36	-9204.7
W|W2,PA2,W1,PA1	10.89	-9401.08
W|W2,SU2,W1,SU1	10.70	-9330.96

Table 6.5: Perplexity of other factored language models

decoded using this language model has a WRA of 91.60%, which is slightly lower than the performance of the normal baseline speech recognition system (91.67%). This might be due to the smoothing technique applied in the development of the language models. Although absolute discounting with the same discounting factor has been applied to both bigram models, the unigram models have been discounted differently. While in the baseline word based language model the unigram models have not been discounted at all, in the equivalent factored model the unigrams have been discounted using Good-Turing discounting technique which is the default discounting technique in SRILM.

The various factored language models (described in 6.1.3) have been used to rescore the lattices and most of them brought a considerable improvement in WRA. The first best factored language model learned by the genetic algorithm outperformed the second best model and the one with fixed backoff (see Table 6.6). The factored language model with fixed backoff did not contribute much to the WRA improvement although it is the best model in terms of the perplexity and the probability it assigns to the test set as it is shown in Table 6.4.

Language models used	Word recognition accuracy in %
Baseline word bigram (FBL)	91.60
FBL + FLM with fixed backoff	91.99
FBL + 1st Best factor combination	92.82
FBL + 2nd Best factor combination	92.55

Table 6.6: WRA improvement with factored language models

All the factored language models that integrate an additional word feature in the n-gram history brought an improvement in WRA. Among the models with two par-

ents, the one that takes POS contributed more to the performance improvement. Although the model in which the probability estimation was conditioned on the previous word and its root has a lower perplexity and assigned higher probability to the test set compared to the others, this model did not achieve the highest improvement in WRA. Models with four parents did not fare better than the ones with two parents, if the maximal n-gram order to be used for transition weight assignment was set to 2. However, when trigrams are used, all the models contributed a notable improvement.

Language models used	Word recognition accuracy in %
Baseline word bigram (FBL)	91.60
FBL + W|W1,POS1	92.87
FBL + W|W1,PR1	92.85
FBL + W|W1,R1	92.75
FBL + W|W1,PA1	92.77
FBL + W|W1,SU1	92.58
FBL + W|W2,POS2,W1,POS1	93.60
FBL + W|W2,PR2,W1,PR1	93.82
FBL + W|W2,R2,W1,R1	93.65
FBL + W|W2,PA2,W1,PA1	93.68
FBL + W|W2,SU2,W1,SU1	93.53

Table 6.7: WRA improvement with other factored language models

Unlike the other factored language models presented so far, root-based language models always caused a degradation of word recognition accuracy (see Table 6.8). Although the higher order root-based model, namely the pentagram, assigned a high probability to the test set, it resulted in a WRA which is below that of the baseline system. This is a surprise because the root-based models have been used together with the baseline model. It may be unobjectionable if no improvement over the baseline has been achieved since a root might represent a set of alternative words and consequently root-based models are less constraining. The results, therefore, indicate that root-based models capture information that is not helpful for the baseline system.

These results also show that a reduction in perplexity of the language models does not always lead to an improvement in WRA.

6.2 Morpheme-based Speech Recognition

In Section 6.1, we have presented the use of morphology-based language models in speech recognition in a lattice rescoring frame work. We showed that most of the

Language models used	Word recognition accuracy in %
Baseline word bigram (FBL)	91.60
FBL + Root bigram	90.77
FBL + Root trigram	90.87
FBL + Root quadrogram	90.99
FBL + Root pentagram	91.14

Table 6.8: WRA with root-based models

morphology-based language models brought an improvement in the performance of a speech recognition system. However, the use of morpheme-based language models in a lattice rescoring framework does not solve the OOV problem. This section presents a morpheme-based speech recognition experiment we have conducted to show how the OOV problem can be reduced by using morphemes as lexical and language modeling units in a speech recognition system.

Most large vocabulary speech recognition systems operate with a finite vocabulary. All the words which are not in the system's vocabulary are considered out-of-vocabulary words. These words are one of the major sources of error in an automatic speech recognition system. When a speech recognition system is confronted with a word which is not in its vocabulary, it may recognize it as a phonetically similar in-vocabulary unit/item. That means the OOV word is mis-recognized. This in turn might cause its neighboring words also to be mis-recognized. Woodland et al. [1995] indicated that each OOV word in the test data contributes to 1.6 errors on the average. Therefore, different approaches have been investigated to cope with the OOV problem and as a consequence to reduce the error rate of automatic speech recognition systems. One of these approaches is vocabulary optimization Bazzi [2002], where the vocabulary is selected in a way that it reduces the OOV rate. This involves either increasing the vocabulary size or including frequent words in a vocabulary. This approach may work for morphologically simple languages like English where a 20k vocabulary has 2% OOV rate and a 65k one has only 0.6% Gales and Woodland [2006].

However, for morphologically rich languages, for which OOV is a severe problem, a much larger vocabulary is required to reach a similarly low OOV rate. Gales and Woodland [2006] indicated that for Russian an 800k and Arabic a 400k vocabulary is required to reduce the OOV rate to 1%. Increasing the vocabulary to alleviate the OOV problem is not the best solution especially for morphologically rich languages as the system complexity increases with the size of the vocabulary. Therefore, modeling sub-word units, particularly morphs, has been used for morphologically rich languages. Prominent work has been reviewed in Section 2.6.1.

For Amharic, the application of automatic word decomposition (using Harris algorithm) for automatic speech recognition has been investigated by Pellegrini and Lamel [2006a]. In their study, the units obtained through decomposition have been used in both lexical and language models. They reported recognition results for four different configurations: full word and three decomposed forms (detaching both prefix and suffix, prefix only and suffix only). A word error rate (WER) reduction over the base line word-based system has been reported using 2 hours of training data in speech recognition in all decomposed forms although the level of improvement varies. The highest improvement (5.2% absolute WER reduction) has been obtained with the system in which only the prefixes have been detached. When both the prefixes and suffixes have been considered, the improvement in performance is small, namely 2.2%. This might be, as the authors indicate, due to the limited span of the n-gram language models.

Decomposing lexical units with the same algorithm led to worse performance when more training data (35 hours)[4] was used [Pellegrini and Lamel, 2007]. This can be explained by a higher acoustic confusability. Pellegrini and Lamel [2007, 2009] tried to solve this problem by using other modified decomposition algorithms. Their starting algorithm was also Morfessor [Creutz and Lagus, 2005] which, however, has been modified by adding different information. In Morfessor baseline, the prior probability of getting N distinct morphs ($p(Lexicon)$) is estimated on the basis of the frequency and length (character sequence probability) of morphs. The first modification made by Pellegrini and Lamel [2007, 2009] affect the calculation of morph length. In Morfessor Baseline character probabilities are static and calculated as a simple ratio of the number of occurrences of the character (irrespective of its place in words) divided by the total number of characters in the corpus. Inspired by Harris' algorithm, Pellegrini and Lamel [2007, 2009] made the calculation context sensitive. The probability that a word beginning (WB[5]) is a morph, is defined as the ratio of the number of distinct letters L(WB) which can follow WB over the total number of distinct letters L. The other modification is adding a phone-based feature in the calculation of $p(Lexicon)$. The third modification is to avoid segmentation if it results in phonetically confusable morphs. During the decomposition process, morphs that differ from each other by only one syllable are compared. If the pair of syllables is among the most frequently confused pairs (found in their previous study [Pellegrini and Lamel, 2006b]), the segmentation is forbidden. They were only able

[4] As indicated by Pellegrini and Lamel [2006a], the speech data are broadcast news data taken from Radio Deutsche Welle (25 hours) and Radio Medhin (12 hours) of which two hours of speech were reserved for development test.

[5] The word beginning symbol WB stands for the strings that begin a given word, from length zero to the length of the word itself.

to achieve a word error rate reduction if the phonetic confusion constraint was used to block the decomposition of words which would result in acoustically confusable units.

In this section, we first show the effect of OOV rate on the performance of an Amharic speech recognition system by using three full-word dictionaries (5k, 20k and 65k). Then, we investigate options to reduce the OOV problem using morphemes as lexical and language modeling units and study its effect on the performance of the system. Since our aim is to study the contribution of morphemes to the reduction of the OOV rate in a speech recognition task and since we wanted to exploit the capability of the recent HTK Decoder (HDecode which allows the use of trigram language models), new sets of acoustic models have been developed instead of using the ones applied in the lattice rescoring experiment. Morfessor [Creutz and Lagus, 2005] has been used to morphologically segment the text corpus required for training and testing in a morpheme-based speech recognition system.

6.2.1 Word-based Recognizers

6.2.1.1 The Speech Corpus

The speech corpus used to develop the speech recognition system is the Amharic read speech corpus described in Section 6.1.1. Again, the 5k development test set has been used for the purpose of this investigation.

6.2.1.2 Acoustic, Lexical and Language Models

The acoustic model consists of 6,610 cross-word triphone HMMs each with 3 emitting states. The states of these models and all the cross-word triphone models that are potentially needed for recognition are tied using decision-tree based state-clustering that reduced the number of triphone models from 77,658 logical models to 10,215 physical ones. Their mixture is added incrementally and 12 Gaussian mixtures have been found to be optimal.

The vocabulary for the three full-word form pronunciation dictionaries has been prepared by taking the most frequent words from the ATC_120k text corpus described in Section 5.3.1 that consists of 120,262 sentences (2,348,150 tokens or 211,120 types). Table 6.9 shows the out-of-vocabulary rates on the 5k development test set that consists of 360 sentences (4,106 tokens or 2,836 distinct words) for these vocabularies. Although we tried to optimize the vocabularies by taking the most frequent words, the OOV rate is still high. The pronunciation dictionaries have then been encoded using the simple procedure described in Section 6.1.1.

Vocabulary	Token OOV (%)	Type OOV (%)
5k	36.43	51.55
20k	20.41	29.23
65k	9.33	13.36

Table 6.9: OOV rate on the 5k development test set

The text corpus from which the vocabularies have been selected has also been used to train the language models. As there are three dictionaries (5k, 20k and 65k), we have developed three trigram language models one for each vocabulary using the SRILM toolkit Stolcke [2002]. The language models are made open by including a special unknown word token. The modified Kneser-Ney smoothing method has been used to smooth all the language models.

6.2.1.3 Performance of Word-based Speech Recognizers

Speech recognition experiments have been performed using the 5k, 20k and the 65k vocabularies. In each case the systems have been evaluated with the 5k development test set. Figure 6.1 presents the word recognition accuracy for each vocabulary. As it can be seen from the figure, the OOV rate decreases when the vocabulary size increases. As the OOV rate decreases the performance of the speech recognition system increases. The best performance (78.3%) has been obtained for the 65k which has OOV rate of 9.33%. The results show that the OOV rate highly affects the performance of a speech recognition system. To deal with this problem, morphemes instead of words have been considered as dictionary entries and units in language models.

6.2.2 Morpheme-based Recognizers

6.2.2.1 Acoustic, Lexical and Language Models

The acoustic model has been developed in a similar fashion as for the word-based recognizers. The training data has a set of 6,459 cross-morph triphone HMMs each with 3 emitting states. The states of these models and all the possible cross-morph triphone models are tied and, therefore, the number of triphone models is reduced from 57,799 logical to 7,685 physical models. Similar to the word-based models, 12 Gaussian mixtures have been found to be optimal.

The ATC_120k corpus has been morphologically segmented using Morfessor. The resulting morphologically segmented text corpus consists of 4,035,656 tokens or 15,925 distinct morphs. In order to facilitate the conversion of morpheme sequences

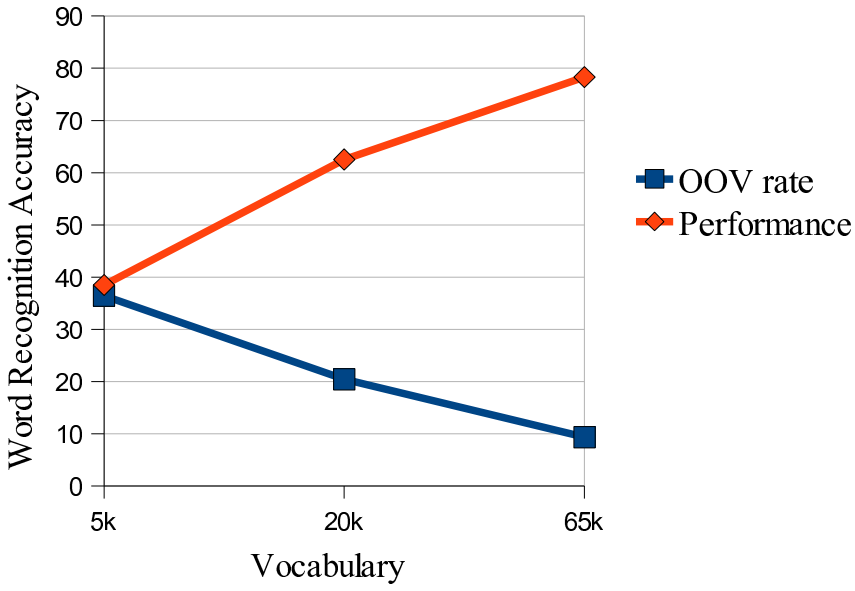

Figure 6.1: Word recognition accuracy of three word-based recognizers

to words, a special word boundary marker has been attached to word boundary morphemes which made the morphemes context-sensitive and consequently increased the number of distinct morphemes to 28,492. From the morphologically segmented corpus, three dictionaries have been prepared: 5k and 20k by taking the most frequent morphs and 28.4k by considering all the morphemes. The morpheme-based OOV rates of these vocabularies on the 5k development test set are presented in Table 6.10 which shows that the OOV rate is highly reduced as a result of using morphs.

Vocabulary	Token OOV (%)	Type OOV (%)
5k	10.75	28.43
20k	0.67	1.83
All (28.4k)	0.03	0.08

Table 6.10: Morph OOV rate of the 5k development test set

In order to analyse the benefit of morpheme-based system interms of covering the OOV words, we calculated word-based OOV rate by counting the number of words which can not be rewriten interms of the morphs found in the vocabularies. Table 6.11 shows the word OOV rate of morpheme-based systems. As it can be observed, the use of morphemes highly reduced the word OOV rate. The token OOV rate for the 5k morph vocabulary, for instance, is reduced by almost 50% from the token OOV rate of the 5k full-word vocabulary (cf. Table 6.9).

As we have three dictionaries (5k, 20k and 28.4k), we have developed three

Vocabulary	Token OOV (%)	Type OOV (%)
5k	18.39	26.06
20k	1.19	1.73
All (28.4k)	0.05	0.07

Table 6.11: Word OOV rate of morph-vocabularies on the 5k development test set

open vocabulary morpheme-based trigram language models, one for each vocabulary. Similar to the word-based language models, the morpheme-based ones have also been smoothed using modified Kneser-Ney smoothing technique.

6.2.2.2 Performance of Morpheme-based Speech Recognizers

The morpheme-based speech recognition system has been evaluated on the 5k development test set using the 5k, 20k and 28.4k morph vocabularies. The results are reported in terms of morph recognition accuracy and word recognition accuracy. The word recognition accuracy has been computed after words have been obtained by concatenating the recognized morph sequence. The best performance (see Table 6.12) has been obtained with the 28.4k morph vocabulary which has an OOV rate of 0.03. Since the OOV rate is very small, an accuracy even higher than the one reported here was expected. The reasons for this disappointing performance (in spite of having a small OOV rate) might be a higher acoustic confusability and the limited language model scope which are peculiar to morpheme-based speech recognition systems.

Vocabulary	Morph Rec. Acc.	Word Rec. Acc.
5k	55.34	50.04
20k	67.67	62.00
28.4k	68.26	62.78

Table 6.12: Performance of morpheme-based speech recognizers

6.2.3 Comparison of Word- and Morpheme-based Speech Recognizers

The morph vocabularies have a very low OOV rate compared to the word vocabularies. This has a positive effect on speech recognition accuracy, especially for small vocabularies, namely 5k. The word-based model has a word recognition accuracy of 38.47% when the 5k vocabulary has been used. On the other hand, the morpheme-based system reaches a word recognition accuracy of 50.04% for the 5k morph vo-

cabulary[6], which means an absolute improvement of 11.57%. However, for the 20k the morpheme-based speech recognizer performed slightly worse (62.00%) than the equivalent word-based system which has a word recognition accuracy of 62.51%. The performance of the recognizer with 28.4k morph vocabulary is only slightly better than the 20k word-based recognizer although it includes all the morphs in the text and has a very low OOV rate. As we have already mentioned, besides the acoustic confusability, the limited scope of the morpheme-based n-gram language model might contribute to the poor performance of the morpheme-based speech recognizer. This has also been commented on by Pellegrini and Lamel [2006a] who suggested the use of higher order n-gram models. Thus, higher order morpheme-based language models have been used in our morpheme-based speech recognizers.

Since it is not possible to use n-gram models higher than trigram in the HTK decoding tool (HDecode), we generated lattices using the 20k and 28.4k vocabulary morpheme-based recognizers and rescored the lattices with a quadrogram morpheme-based language model which has been developed in the same manner as the trigram models. As it can be seen from Table 6.13, a 1% absolute word recognition accuracy improvement (over the 20k word-based recognizer) has been obtained for the 20k vocabulary morpheme-based recognizers as a result of rescoring the lattices with a quadrogram language model. However, the performance of the 28.4k which has a very low OOV rate (0.03) and covers all the morphemes in the text is still smaller than the performance of the 65k word-based recognizer (78.3%) which has a considerably higher OOV rate (9.33) although the result is better than the one that uses a trigram morpheme-based language model. This might indicate that acoustic confusability is the most influencial factor for the performance degradation of a morpheme-based speech recognizer on large vocabularies.

Vocabulary	Morph Rec. Acc.	Word Rec. Acc.
20k	68.92	63.51
28.4k	69.70	64.46

Table 6.13: Lattice rescoring with a quadrogram morpheme-based language model

As it can be seen from Table 6.14, rescoring with a pentagram language model did not lead to further improvement. Rather, the morph and word recognition accuracies (for both 20k and 28.4k vocabularies) became worse than the recognizer that used

[6]Comparing the morpheme-based systems directly with the word-based ones may not be fair because they have a higher coverage than word-based systems of the same vocabulary size. On the other hand, the morpheme-based systems are also dis-favoured by the concatenation of illegal morph-sequences, increasing number of small and acoustically confusable units and a limited language model scope.

the quadrogram morpheme-based language model. This is similar with that of Diehl et al. [2009] who obtained notable reduction in word error rate using quadrogram language models but, only minimal or no improvement (over the systems that use qadrogram language models) using pentagram models. Our result might be due to data sparseness. As the language model training corpus is very small, many of the pentagrams might not have been encountered in the training data and, therefore, are estimated in terms of lower order n-grams. Regarding the language model quality, the pentagram language models did not lead to much perplexity improvement (less than 1%) compared to the quadrogram ones for the 20k and the 28.4k vocabularies. The perplexity gains of the quadrogram language models over the trigram ones are 8.291% and 8.386% for the 20k and 28.4k vocabularies, respectively.

Vocabulary	Morph Rec. Acc.	Word Rec. Acc.
20k	67.69	62.17
28.4k	68.48	63.17

Table 6.14: Lattice rescoring with a pentagram morpheme-based language model

Chapter 7

Conclusion and Recommendation

7.1 Introduction

The purpose of this research was to explore the best way of modeling the Amharic language using morphemes as units. To this end, we have developed various morphology-based (morpheme-based and factored) language models by which we demonstrated the possibility of using morphological information in Amharic language modeling. This chapter presents our conclusive remarks and recommendations for future work.

7.2 Conclusion

Language models have been developed using both statistical and linguistic morphemes. The results of our experiments have shown that linguistic morpheme-based language models are better than the statistical morph-based ones. In linguistic morph-based language models, the root-pattern morphological phenomenon has been modeled explicitly. Moreover, the segmentation quality is certainly better than the one provided by Morfessor. The results also have shown that an explicit treatment of the root-pattern morphological phenomenon is superior than simply ignoring it. These models have been compared with word-based ones on the basis of the probability they assigned to the test set and it has been found that word based language models fared better.

However, since the best way of determining the quality of language models is to apply them to the specific application for which they have been designed and see whether they contribute an improvement or not, speech recognition experiments have been conducted in a lattice rescoring framework where word lattices have been rescored with interpolated models (word and morph-based). All morpheme-based language models improved word recognition accuracy. Based on the lattice rescoring experiment alone, we were not able to conclude whether the word based language models are superior, since interpolated models have been used in rescoring. It is possible, however, to conclude that the use of morpheme-based language models as a complementary tool (to the word based language models) is fruitful for speech

recognition systems. Moreover, the morph-based language models provide the advantage of reducing OOV words rate which is a serious problem in morphologically rich languages.

Using morpheme-based language models in a lattice rescoring framework does not solve the OOV problem. Therefore, speech recognition experiments for Amharic have been conducted to study the effect of OOV words problem and to find out whether the problem can be reduced by using morphemes as lexical and language model units. For the word-based systems, the OOV rate decreases as the vocabulary size increases and word recognition accuracy increases as the OOV rate decreases. Using morphemes as dictionary entries and language model units highly reduced the OOV rate and consequently boosted the word recognition accuracy, especially for small vocabularies (5k). However, as the morph vocabulary grows, the performance of morpheme-based speech recognition was not as expected since it suffers from acoustic confusability and limited n-gram language model scope. The word recognition accuracy improvement obtained as a result of lattice rescoring with higher order morpheme-based language model (quadrogram) shows convincingly that the morpheme-based speech recognizer suffers from limited n-gram language model scope. But the acoustic confusability is more important.

Exploiting the nature of Amharic roots, we proposed and developed root-based language models (in the framework of factored language modeling) as a solution for the loss of word-level dependencies. The root-based language models have high test set probability. However, since a root might stand for a set of words, the root-based language models are less constraining. Therefore, another speech recognition experiment has been conducted to analyse their contribution to the improvement of speech recognition performance. The results showed that the root-based language models did not contribute to the improvement of word recognition accuracy. The use of root-based language models rather degraded the performance of the baseline system.

Since factored language modeling makes possible to integrate any kind of relevant information to language models, several factored language models have also been developed. It was possible to gain a reduction in perplexity values and all the models that take an extra information were better than the word only models. This result received further support from the word recognition accuracy improvement gained as a result of using these models in a speech recognition task. This enables us to conclude that integrating morphological information into word-based models leads to better quality language models.

It has also been found that Kneser-Ney smoothing technique and its variations consistently outperformed alternative smoothing techniques irrespective of the units

used in language modeling.

7.3 Recommendations

Based on the findings of our research and the knowledge obtained from the literature, the following recommendations are forwarded for future work.

Our experiments showed that linguistic morphemes resulted in better quality language models although the amount of data used for the experiment is limited due to the laborious task of manual segmentation. We believe that developing linguistic morph-based language models with more data will result in more robust language models. For this, however, there is a need for a rule-based Amharic morphological analyzer. We recommend to further develop the finite state based morphological analyzer of Saba and Gibbon [2005] by increasing the coverage of the dictionaries and modifying the format of the output. Extending the coverage of the dictionaries can be done by extracting words from electronic dictionaries, from the manually tagged corpus [Girma and Mesfin, 2006] or by tagging texts using automatic methods. It might even be possible to generate the entries of a root dictionary automatically from Amharic consonants using a combinatorial approach. Since this method might also generate invalid roots, using the phonological rules of the language as additional constraints (to avoid invalid roots) is advisable. For morpheme-based language modeling, the output of the system should be the constituent morphemes of a given word instead of the morphological features. Thus, the format of the output of the morphological analyzer should also be modified accordingly. As the Amharic language is under-resourced, it would even be advantageous to provide alternative output formats so that the morphological analyzer can also be applied to other natural language processing tasks. We also recommend to explore the functionalities of the recently developed Amahric morphological analyser, namely HornMorpho [Gasser, 2010a], for future use.

Although the morpheme-based recognizer benefits from the low OOV rate, it suffers from the small, acoustically confusable units and the limited span of the n-gram language model. We used lattice rescoring with higher order n-grams and obtained an improvement in word recognition accuracy. Further improvement in morpheme-based speech recognition can be expected if care is taken (for instance, using confusion constraints as in Pellegrini and Lamel [2009]) to avoid acoustically confusable units during or after morphological segmentation. Moreover, we just concatenated the recognized morpheme sequences up to a word boundary marker and no effort has been made to avoid concatenation of illegal morpheme sequences. Attempts in this line may also improve the performance of morpheme-based speech

recognizer where a morpheme-based language model is one of the main components. For example, rules (such as *ignore the subject marker morph if it comes at the beginning of a morph sequence*) could help to avoid the concatenation of illegal morph sequences.

We developed root-based language models to compensate the loss of word level dependencies. Although these models assigned high probability to the test set, no benefit has been gained from them in improving the performance of a speech recognition system. Improvement of these models by adding other word features, but still maintaining word level dependencies is recommended. Moreover, since dynamic cache language models overcome the limitation of n-gram models in modeling dependencies longer than n, the development of morph-based dynamic cache models to capture word level dependencies is worth exploring.

We applied the morpheme-based language models to an automatic speech recognition task. The morpheme-based language models can also be applied to other natural language applications. Now a days, morphological decomposition, and consequently morpheme-based language model is used in statistical machine translation into morphologically complex languages [Labaka et al., 2007, Badr et al., 2008, Oflazer, 2008]. We, therefore, specifically recommend the use of morph-based language models for Amharic statistical machine translation since the problem of acoustic confusability is not an issue in statistical machine translation.

Last, but not least, we recommend the development of text processing tools, such as a spelling checker, for Amharic since the availability of such tools will facilitate research in Amharic natural language processing.

Appendix A

IPA Representation of Amharic Sounds

A.1 Consonants

	Bilabial	Labio dental	Alveolar	Post alveolar	Palatal	Velar	Glotal	Labialized Velar
Plosive	p 'ፐ' b 'ቡ'		t 'ት' d 'ድ'			k 'ክ' g 'ግ'	? 'እ'	k^w 'ኩ' g^w 'ጉ'
Affricate				tʃ (č) 'ች' dʒ (ǧ) 'ጅ'				
Nasal	m 'ም'		n 'ን'		ɲ (ñ) 'ኝ'			
Fricative		f 'ፍ'	s 'ስ' z 'ዝ'	ʃ (š) 'ሽ' ʒ (ž) 'ዥ'			h 'ህ'	
Tap/Trill			r 'ር'					
Approximant	w 'ው'				j (y) 'ይ'			
Lateral Approximant			l 'ል'					
Ejective Stop	(p') 'ጵ'		t' 'ጥ'			k' 'ቅ'		k^w' 'ኵ'
Ejective Affricate				tʃ' (Č) 'ጭ'				
Ejective Fricative			s' 'ፅ'					

Table A.1: IPA Representation of Amharic consonants

A.2 Vowels

	front	center	back
high	i 'ኢ'	I 'እ'	u 'ኡ'
mid	e 'ኤ'	ɨ (ä) 'ኧ'	o 'ኦ'
low		a 'አ'	

Table A.2: IPA Representation of Amharic vowels

Bibliography

Solomon Teferra Abate. *Automatic Speech Recognition for Amharic*. PhD thesis, Univ. of Hamburg, 2006. 4, 5, 69, 75, 98, 99

Solomon Teferra Abate, Wolfgang Menzel, and Bairu Tafila. An Amharic speech corpus for large vocabulary continuous speech recognition. In *Proceedings of 9th. European Conffference on Speech Communication and Technology, Interspeech-2005*, 2005. 66, 67, 76, 98

Sisay Fissaha Adafre. *Adding Amharic to a Unification-Based Machin Translation System*. Peter Lang, Frankfurt am Main, 2004. 37

Sisay Fissaha Adafre. Part of speech tagging for Amharic using conditional random fields. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 47–54, 2005. 62, 68, 83, 84

Nega Alemayehu and Peter Willett. Stemming of amharic words for information retrieval. *Literary and Linguistic Computing*, 17(1):1–17, 2002. 60, 61, 68

Nega Alemayehu and Peter Willett. The effectiveness of stemming for information retrieval in amharic. *Program: electronic library and information system*, 37(4): 254–259, 2003. 61

Atelach Alemu, Lars Asker, and Mesfin Getachew. Natural language processing for amharic: Overview and suggestions for a way forward. In *Proceeding of TALN-2003 Workshop on Natural Language Processing of Minority Languages and Small Languages*, 2003. 3

Mengistu Amberber. *Verb Classes and Transitivity in Amharic*. LINCOM EUROPA, Muenchen, 2002. 38, 39

Saba Amsalu and Dafydd Gibbon. Finite state morphology of amharic. In *Proceedings of International Conference on Recent Advances in Natural Language Processing*, pages 47–51, 2005. 62, 68, 115

Atelach Alemu Argaw and Lars Asker. An amharic sremmer: Reducing words to their citation forms. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources - ACL 2007*, pages 104–110, 2007. 62, 68

Scott Elliot Axelrod, Peter Andreas Olsen, Harry William Printz, and Peter Vicent de Souza. Determining and using acoustic confusablity, acoustic perplexity and synthetic acoustic word error rate, 2007. URL `http://www.freepatentsonline.com/7219056.pdf`. 14

Ibrahim Badr, Rabih Zbib, and James Glass. Segmentation for english-to-arabic statistical machine translation. In *Proceedings of ACL-08, HLT short paper*, pages 153–156, 2008. 116

Lalit R. Bahl and Robert L. Mercer. Part-of-speech assignment by a statistical decision algorithm. In *IEEE International Symposium on Information Theory*, pages 88–89, 1976. 2

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190, 1983. 1

Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. A tree-based statistical language model for natural language speech recognition. In *IEEE Transactions on Acoustic, Speech, and Signal Processing*, volume 37, pages 1001–1008, 1989. 23

Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the HLT-NAACL Conference*, pages 113–120, 2004. 2

Abiyot Bayou. Developing automatic word parser for amharic verbs and their derivation. Master's thesis, Addis Ababa University, 2000. 60, 68

Tesfaye Bayu. Automatic morphological analyzer for amharic: An experiment employing unsupervised learning and autosegmental analysis approaches. Master's thesis, Addis Ababa University, 2002. 61, 68

I. Bazzi. *Modelling Out-of-Vocabulary Words for Robust Speech Recognition*. PhD thesis, Massachsetts Institute of Technology, 2002. 105

M. L. Bender and H. Fulass. *Amharic Verb Morphology: A Generative Approach*. Michigan State University, Michigan, 1978. 35, 36, 38, 79

M.L. Bender, J.D. Bowen, R.L. Cooper, and C.A. Ferguson. *Languages in Ethiopia*. Oxford Univ. Press, London, 1976. 33, 34

Solomon Berhanu. Isolated amharic consonant-vowel (CV) syllable recognition: An experiment using the hidden markov model. Master's thesis, Addis Ababa University, 2001. 4

Jayadev Billa, Kristine Ma, John W. McDonough, George Zavaliagkos, David R. Miller, Kenneth N. Ross, and Amro El-Jaroudi. Multilingual speech recognition: The 1996 byblos callhome system. In *Proceedings of Eurospeech*, pages 363–366, 1997. 29

Thorsten Brants. TnT — a statistical part-of-speech tagger. In *Proceedings of the 6th ANLP*, 2000. 86

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. 2

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992. 1, 22

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993. 2

W. Byrne, J. Hajič, P. Ircing, F. Jelinek, S. Khudanpur, P. Krebc, and J. Psutka. On large vocabulary continuous speech recognition of highly inflectional language - czech. In *Proceeding of the European Conference on Speech Communication and Technology*, pages 487–489, 2001. 27

William Byrne, Jan Hajič, Pavel Ircing, Pavel Krbec, and Josef Psutka. Morpheme based language models for speech recognition of czech. In *Proceeding of International Conference on Text, Speech and Dialogue*, pages 139–162, 2000. 25, 26

Kenan Carki, Petra Geutner, and Tanja Schultz. Turkish lvcsr: towards better speech recognition for agglutinative languages. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3:1563–1566, 2000. 25, 26

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, Massachusetts, 1998. 14, 17, 19, 20, 21, 73

Ghinwa Choueiter, Daniel Povey, Stanley F. Chen, and Geoffrey Zweig. Morpheme-based language modeling for arabic lvcsr. *Proceedings of ICASSP*, 1, 2006. 25, 31

Kenneth W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied Natural Language Processing*, pages 136–143, 1988. 2

Kenneth W. Church and William A. Gale. Probability scoring for spelling correction. *Statistics and Computing*, 1(2):93–103, 1991. 2

Ronald Cole, Joseph Mariani, Hans Uszkoreit, Giovanni Battista Varile, Annie Zaenen, Antonio Zampolli, and Victor Zue, editors. *Survey of the State of the Art in Human Language Technology*. Cambridge University Press and Giardini, 1997. 1

Mathias Creutz. *Induction of the Morphology of Natural language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition*. PhD thesis, Helsinki University of Technology, 2006. 56, 57, 58, 59, 60, 69, 70, 72

Mathias Creutz and Krista Lagus. Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.1. Technical Report A81, Neural Networks Research Center, Helsinki University of Technology, 2005. 28, 55, 56, 68, 72, 99, 106, 107

Mathias Creutz and Krister Lindén. Morpheme segmentation gold standards for finnish and english. Technical Report A77, Helsinki University of Technology, 2004. 59, 70

Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. Analysis of morph-based speech recognition and the modeling of out-of-vocabulary words across languages. In *Proceedings of NAACL HLT 2007*, pages 380–387, 2007. 29

W. B. Croft and John Lafferty, editors. *Language Modeling for Information Retrieval*. Kluwer Academic, Amsterdam, 2003. 2

Béatrice Daille, Cécile Fabre, and Pascale Sébillot. *Application of Computational Morphology*, pages 210–234. Cascadilla Press, Somerville, 2002. 51

C. H. Dawkin. *The Fundamentals of Amharic*. Sudan Interior Mission, Addis Ababa, 1960. 35, 36

Girma Awgichew Demeke and Mesfin Getachew. Manual annotation of Amharic news items with part-of-speech tags and its challenges. *ELRC Working Papers*, II(1), 2006. xiii, 84, 85, 115

A. Demoz. *The Meaning of Some Derived Verbal Stems in Amharic*. PhD thesis, Univ. of California, 1964. 38

F. Diehl, M.J.F. Gales, M. Tomalin, and P.C. Woodland. Morphological analysis and decomposition for arabic speech-to-text systems. In *Proceedings of INTERSPEECH 2009*, 2009. 112

Kevin Duh and Katrin Kirchhoff. Automatic learning of language model structure. In *Proceeding of International Conference on Computational Linguistics*, 2004. 32, 93, 94, 102

Abdessamad Echihabi and Daniel Marcu. A noisy-channel approach to question answering. In *Proceedings of the 41st Anual Meeting on Association for Computational Linguistics*, volume 1, pages 16–23, 2003. 2

Amr El-Desoky, Christian Gollan, David Rybach, Ralf Schlüter, and Hermann Ney. Investigating the use of morphological decomposition and diacritization for improving arabic lvcsr. In *Proceedings of INTERSPEECH 2009*, 2009. 25, 29

Marco Ferretti, Glullo Maltese, and Stefano Scarci. Language model and acoustic model information in probablistic speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing - ICASSP-89*, pages 707–710, 1989. 14

Sisay Fissaha and Johann Haller. Amharic verb lexicon in the context of machine translation. In *Proceedings of the 10th Conference on Traitement Automatique des Langues Naturelles*, volume 2, pages 183–192, 2003. 61, 68

William A. Gale and Geoffrey Sampson. Good-turing frequency estimation without tears. *Quantitative Linguistics*, 2:217–237, 1995. 16

M. Gales and P. Woodland. Recent progress in large vocabulary continuous speech recognition: An htk perspective, 2006. URL `http://svr-www.eng.cam.ac.uk/~mjfg/icassp06_tutorial.pdf`. 105

Björn Gambäck, Fredrik Olsson, Atelach Alemu Argaw, and Lars Asker. Methods for Amharic part-of-speech tagging. In *Proceedings of the EACL Workshop on Language Technologies for African Languages - AfLaT 2009*, pages 104–111, March 2009. 83, 84, 85, 87, 88

Michael Gasser. Hornmorpho 2.1 user's guide, 2010a. URL http://www.cs.indiana.edu/~gasser/L3/horn2.1.pdf. 63, 115

Michael Gasser. A dependency grammar for amharic. In *Proceedings of the Workshop on Language Resources and Human Language Technologies for Semitic Languages*, pages 12–18, 2010b. 63

J. L. Gauvain, L. Lamel, G. Adda, and D. Matrouf. The limsi 1995 hub3 system. In *Proceedings of DARPA Spoken Language Technology Workshop*, pages 105–111, 1995. 13

Mesfin Getachew. Automatic part of speech tagging for Amharic language: An experiment using stochastic hmm. Master's thesis, Addis Ababa University, 2000. 83, 84

P. Geutner. Using morphology towards better large-vocabulary speech recognition systems. In *Proceedings of IEEE International on Acoustics, Speech and Signal Processing*, volume I, pages 445–448, 1995. 5, 25, 26

Jesús Giménez and Lluís Màrquez. Svmtool: A general pos tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004. 86, 88

Molalgne Girmaw. An automatic speech recognition system for amharic. Master's thesis, Royal Institute of Technology, 2004. 4

John Goldsmith. Linguistica: An automatic morphological analyzer. In *Proceedings of the Main Session of the Chicago Linguistic Society's 36th meeting*, volume 36-1, 2000. 55, 56, 61, 68

I. J. Good. The population frequencies of species and the esstimation of population parameters. *Biometrika*, 40(3/4):237–264, 1953. 16

Joshua T. Goodman. A bit of progress in languge modeling: Extended version. Technical Report MSR-TR-2001-72, Machine Learning and Applied Statistics Group, Micrososft Research, 2001. 23

Kadri Hacioglu, Bryan Pellom, Tolga Ciloglu, Ozlem Ozturk, Mikko Kurimo, and Mathias Creutz. On lexicon creation for turkish lvcsr. In *Proceedings of the 8th European Conference on Speech Communication and Technology - Eurospeech 2003*, pages 1165–1168, 2003. 13

Margaret A Hafer and Stephen F. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11-12):371–385, 1974. 55

Zellig S. Harris. From phoneme to morpheme. *Language*, 31(2):190–222, 1955. 55

Ilana Heintz. *Arabic Language Modeling with Stem-Derived Morphemes for Automatic Speech Recognition*. PhD thesis, Graduate Program in Linguistics, The Ohio State University, 2010. 25, 30

Teemu Hirsimäki, Mathias Creutz, Vesa Siivola, and Mikko Kurimo. Morphologically motivated language models in speech recognition. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, pages 121–126, 2005. 5, 25, 28, 60

Xuedong Huang, Fileno Alleva, Hsiao wuen Hon, Mei yuh Hwang, and Ronald Rosenfeld. The sphinx-ii speech recognition system: An overview. *Computer, Speech and Language*, 7:137–148. 23

Rukmini Iyer, Mari Ostendorf, and J. Robin Rohlicek. Language modeling with sentence-level mixtures. In *Proceedings of the workshop on Human Language Technology HLT '94*, pages 82–87, 1994. 24

Rukmini M. Iyer and Mari Ostendorf. Modeling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39, 1999. 24

F. Jelinek. *Self-organized Language Modeling for Speech Recognition*, pages 450–506. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1990. 2, 4

F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss. A dynamic language model for speech recognition. In *Proceedings of the workshop on Speech and Natural Language HLT '91*, pages 293–295, 1991. 24

Frederick Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge,London, 1997. 10

Frederick Jelinek and Robert L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, 1980. 17

Jean-Claude Junqua and Jean-Paul Haton. *Robustness in Automatic Speech Recognition: Fundamentals and Applications*. Kluwer Academic, London, 1996. 2

Daniel S. Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, New Jersey, 2nd. ed. edition, 2008. xv, 1, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 24, 56, 83

Lauri Karttunen. Constructing lexical transducers. In *The 15th International Conference on Computational Linguistics - COLING'94*, pages 406–411, 1994. 54

Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, 1987. 18

Dimitar Kazakov and Suresh Manandhar. Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming. *Machine Learning*, 43(1-2):121–162, 2001. 53, 55

Mark D. Keringhan, Kenneth W. Church, and William A. Gale. A spelling correction program based on a noisy channel model. In *Proceedings of the 13th International Conference on Computational Linguistics*, volume 2, pages 205–210, 1990. 2

Daniel Kiecza and Tanja Schultz Alex Waibel. Data-driven determination of appropriate dictionary units for korean lvcsr. In *In Proceedings of ICASSP*, pages 323–327, 1999. 25, 30, 31

Katrin Kirchhoff, Jeff Bilmes, John Henderson, Richard Schwartz, Mohamed Noamany, Pat Schone, Gang Ji, Sourin Das, Melissa Egan, Feng He, Dimitra Vergyri, Daben Liu, and Nicolae Duta. Novel speech recognition models for arabic. Technical report, Johns-Hopkins University Summer Research Workshop, 2002. 25, 29, 30, 31, 98

Katrin Kirchhoff, Jeff Bilmes, Sourin Das, Nicolae Duta, Melissa Egan, Gang Ji, Feng He, John Henderson, Daben Liu, Mohamed Noamany, Pat Schone, Richard Schwartz, and Dimitra Vergyri. Novel approaches to Arabic speech recognition: Report from the 2002 johns-hopkins summer workshop. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 344–347, 2003. 32, 83

Katrin Kirchhoff, Dimitra Vergyri, Jeff Bilmes, Kevin Duh, and Andreas Stolcke. Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech and Language*, 20(4):589–608, 2006. 30, 94

Katrin Kirchhoff, Jeff Bilmes, and kevin Duh. Factored language models - a tutorial. Technical report, Dept. of Electrical Eng., Univ. of Washington, 2008. 32

Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1): 91–107, 2002. 2

Okan Kolak and Philip Resnik. Ocr error correction using a noisy channel model. In *Proceedings of the second International Conference on Human Language Technology Research*, pages 257–262, 2002. 2

G. E. Kopec and P. A. Chou. Document image decoding using markov source models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6): 602–617, 1994. 2

Kimmo Koskenniemi. *Two-level Morphology: A General Computaional Model for Word-form Recognition and Production*. PhD thesis, University of Helsinki, 1983. 54

Roland Kuhn. Speech recognition and the frequency of recently used words: a modified markov model for natural language. In *Proceedings of the 12th conference on Computational linguistics*, pages 348–350, 1988. 24

Ronald Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583, 1990. 24

Ronald Kuhn and Renato De Mori. Corrections to "a cache-based language model for speech recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):691–692, 1992. 24

Julien Kupiec. Probabilistic models of short and long distance word dependencies in running text. In *Proceedings of the workshop on Speech and Natural Language HLT '89*, pages 290–295, 1989. 24

Oh-Wook Kwon. Performance of lvcsr with morpheme-based and syllable-based recognition units. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 3:1567–1570, 2000. 25, 31

Gorka Labaka, Nicolas Stroppa, Andy Way, and Kepa Sarasola. Comparing rule-based and data-driven approaches to spanish-to-banque machine translation. In *Proceedings of Machine Translation Summit XI*, 2007. 116

Raymond Lau, Ronald Rosenfeld, and Salim Roukos. Trigger-based language models: A maximum entropy approach. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 45–48, 1993. 22, 24

Wolf Leslau. *Introductory Grammar of Amharic*. Harrassowitz Verlag, Wiesbaden, 2000. xiii, 35, 40, 41, 48

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge,London, 1999. 14, 15, 16, 19, 67

Sven Martin, Christoph Hamacher, Jörg Liermann, Frank Wessel, and Hermann Ney. Assessment of smoothing methods and complex stochastic language modeling. In *In Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 1939–1942, 1999. 23

Beáta Megyesi. Comparing data-driven learning algorithms for pos tagging of Swedish. In *Proceedings of the 2001 Conference on Emperical Methods in Natural Language Processing*, pages 151–158, 2001. 86

Hermann Ney and Ute Essen. On smoothing techniques for bigram-based natural language modelling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP-91*, pages 825–828, 1991. 19

Hermann Ney, Ute Essen, and Reinhard Kneser. On the estimation of 'small' probablities by leaving-one-out. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1202–1212, 1995. 19, 20, 21

Thomas Niesler. *Category-based Statstical Language Models*. PhD thesis, University of Cambridge, 1997. 22

Kemal Oflazer. Statistical machine translation into a morphologically complex language. In *Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing CICLing'08*, pages 376–387, 2008. 116

Thomas Pellegrini and Lori Lamel. Investigating automatic decomposition for asr in less represented languages. In *Proceedings of INTERSPEECH 2006*, 2006a. 94, 106, 111

Thomas Pellegrini and Lori Lamel. Experimental detection of vowel pronunciation variants in amharic. In *Proceedings of LREC*, 2006b. 106

Thomas Pellegrini and Lori Lamel. Using phonetic features in unsupervised word decompounding for asr with application to a less-represented language. In *Proceedings of INTERSPEECH 2007*, pages 1797–1800, 2007. 106

Thomas Pellegrini and Lori Lamel. Automatic word decompounding for asr in a morphologically rich language: Application to amharic. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):863–873, 2009. 106, 115

Eric Sven Ristad. A natural law of succession, 1995. 15

Ronald Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellown University, 1994. 23

Ronald Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 10:187–228, 1996. 24

Ronald Rosenfeld. Statistical language modeling and n-grams, 1997. URL `http://www.cs.cmu.edu/afs/cs/academic/class/11761-s97/WWW/tex/Ngrams.ps`. 12

Ronald Rosenfeld. Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000. 13, 14, 23

Salim Roukos. *Language Representation*. Cambridge University Press and Giardini, 1997. 1

Hussien Seid and Björn Gambäck. A speaker independent continuous speech recognizer for amharic. In *Proceeding of INTERSPEECH-2005,9th European Conference on Speech Communication and technology*, 2005. 4

Zegaye Seifu. HMM based large vocabulary, speaker independent, continuous amharic speech recognizer. Master's thesis, Addis Ababa University, 2003. 4

Vesa Siivola, Teemu HirsimÃďki, Mathias Creutz, and Mikko Kurimo. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In *Proceedings of Eurospeech*, pages 2293–2296, 2003. 25, 28, 29, 30, 60

Abdelhadi Soudi, Günter Neumann, and Antal van den Bosch. *Arabic Computational Morphology: Knowledge-based and Emperical Methods*, pages 3–14. Springer, Dordrecht, The Netherlands, 2007. 54

Richard Sproat. *Morphology and Computation*. The MIT Press, London, 1992. 5, 51, 53

Andreas Stolcke. SRILM — an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*, volume II, pages 901–904, 2002. 66, 74, 99, 108

Martha Yifiru Tachbelie. Application of amharic speech recognition system to command and control computer: An experiment with microsoft word. Master's thesis, Addis Ababa University, 2003. 4

Martha Yifiru Tachbelie and Wolfgang Menzel. *Morpheme-based Language Modeling for Inflectional Language – Amharic*. John Benjamin's Publishing, Amsterdam and Philadelphia, 2009. 82

Martha Yifiru Tachbelie, Solomon Teferra Abate, and Wolfgang Menzel. Morpheme-based language modeling for amharic speech recognition. In *Proceedings of the 4th Language and Technology Conference*, pages 114–118, 2009. 101

Kinfe Tadesse. Sub-word based amharic word recognition: An experiment using hidden markov model (HMM). Master's thesis, Addis Ababa University, 2002. 4

Anbessa Teferra and Grover Hudson. *Essentials of Amharic*. Köppe, Köln, 2007. 33, 34, 38

Saba Amsalu Teserra. *Bilingual Word and Chunk Alignment: A Hybrid System for Amharic and English*. PhD thesis, Universität Bielefeld, 2007. 62

E. G. Titov. *The Modern Amharic Language*. NAUKA Publishing House, Moscow, 1976. xiii, 40, 47, 48

Harald Trost. *Computational Morphology*, pages 25–47. Oxford University Press, Oxford, 2003. 51, 54

Antal van den Bosch. *Learning to pronounce Written words: A study in Inductive Language Learning*. PhD thesis, University of Maastricht, 1997. 55

Dimitra Vergyri, Katrin Kirchhoff, Kevin Duh, and Andreas Stolcke. Morphology-based language modeling for Arabic speech recognition. In *Proceedings of International Conference on Spoken Language Processing*, pages 2245–2248, 2004. 5

R. M. Voigt. The classification of central semitic. *Journal of Semitic Studies*, (32): 1–21, 1987. 33

E. Whittaker and P. Woodland. Particle-based language modeling. In *Proceeding of International Conference on Spoken Language Processing*, pages 170–173, 2000. 25, 27, 29, 98

E. W. D. Whittaker, J. M. Van Thong, and P. J. Moreno. Vocabulary independent speech recognition using particles. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 315–318, 2001. 25, 28

Richard Wicentowski. Multilingual noise-robust supervised morphological analysis using the wordframe model. In *Proceedings of the Workshop of the ACL Special*

Interest Group on Computational Pholonolgy and Morphology, pages 70–77, 2004. 55

Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 34(3):1085–1094, 1991. 18

P. C. Woodland, C. J. Leggetter, J. J. Odell V. Valtchev, and S. J. Young. The 1994 HTK large vocabulary speech recognition system. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 73–76, 1995. 105

Baye Yimam. Root reduction and extensions in amharic. *Ethiopian Journal of Language and Literature*, (9):56–88, 1999. 35, 36, 39, 40

Baye Yimam. *yäamarIña säwasäw*. EMPDE, Addis Ababa, 2nd. ed. edition, 2000EC. 4, 33, 41, 44, 45, 47, 79

Baye Yimam. The interaction of tense, aspect, and agreement in amharic syntax. In *Proceedings of the 35th Annual Conference on African Linguistics*, pages 193–202, 2006. 45

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Cambridge University Engineering Department, 2006. 3, 75

Petr Zemánek. Clara (corpus linguae arabicae: An overview). In *Proceedings of ACL/EACL 2001 Workshop on Arabic Language Processing: Status and Prospects*, 2001. 67

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbst verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Hamburg, im August 2010

Martha Yifiru Tachbelie