Sub-word Based Language Modeling for Amharic

Martha Yifiru Tachbelie, Wolfgang Menzel
Department of Informatik, University of Hamburg, Germany
{tachbeli, menzel}@informatik.uni-hamburg.de

Abstract

This paper presents sub-word based language models for Amharic, a morphologically rich and under-resourced language. The language models have been developed (using an open source language modeling toolkit - SRILM) with different n-gram order (2 to 5) and smoothing techniques. Among the developed models, the best performing one is a 5gram model with modified Kneser-Ney smoothing and with interpolation of n-gram probability estimates.

Keywords

Language modeling, sub-word based language modeling, morph-based language modeling, Amharic.

1.Introduction

1.1. Amharic Word Morphology

Amharic is a member of the Ethio-Semitic languages, which belong to the Semitic branch of the Afro-Asiatic super family [23]. It is related to Hebrew, Arabic, and Syrian. Amharic is a major language spoken mainly in Ethiopia. According to the 1998 census, it is spoken by over 17 million people as a first language and by over 5 million as second language throughout different regions of Ethiopia. Amharic is also spoken in other countries such as Egypt and Israel [4].

Like other Semitic languages such as Arabic, Amharic exhibits a root-pattern morphological phenomenon. A root is a set of consonants (called radicals) which has a basic 'lexical' meaning. A pattern consists of a set of vowels which are inserted (intercalated) among the consonants of the root to form a stem. The pattern is combined with a particular prefix or suffix to make a single grammatical form [3] or to form another stem [2]. For example, the Amharic root sbr means 'break', when we intercalate the pattern ä-ä and attach the suffix ä we get säbbärä[1] 'he broke' which is the first form of a verb (3rd person masculine singular in past tense as in other semitic languages) [3]. In addition to this non-concatenative morphological feature, Amharic uses different affixes to form inflectional and derivational word forms.

Some adverbs can be derived from adjectives but, adverbs are not inflected. Nouns are derived from other basic nouns, adjectives, stems, roots, and the infinitive form of a verb by affixation and intercalation. For example, from the noun ləğ 'child' another noun ləğnät 'childhood'; from the adjective däg 'generous' the noun dägnät 'generosity'; from the stem sənəf, the noun sənəfna 'laziness'; from root qld, the noun qäləd 'joke'; from infinitive verb mäsəbär 'to break' the noun mäsəbäriya 'an instrument used for breaking' can be derived.

Case, number, definiteness, and gender marker affixes inflect nouns. Table 1 presents, as an example, the genitive case markers that inflect nouns.

Table 1. Genitive case markers (Adapted from [21])

Person	Singular		Plural
	Vowel ending	Consonant ending	
1st	-ye	-e	-ačč*n*
2nd masculine	-h	-ih	-ačč*hu*
2nd feminine	-š	-iš	
2nd polite	-wo	-wo	
3rd masculine	-w	-u	-ačč*äw*
3rd feminine	-wa	-wa	
3rd polite	-ačč*äw*	-ačč*äw*	

Adjectives are derived from nouns, stems or verbal roots by adding a prefix or a suffix. For example, it is possible to derive dənəgayama 'rocky' from the noun dənəgay 'rock, stone'; zənəgu 'forgetful' from the stem zənəg; sänäf 'lazy' from the root s_n_f by suffixation and intercalation. Adjectives can also be formed through compounding. For instance, hodäsäfi 'tolerant, patient', is derived by compounding the noun hod 'stomach' and the adjective säfi 'wide'. Like nouns, adjectives are inflected for gender, number, and case [2].

Unlike the other word categories such as noun and adjectives, the derivation of verbs from other parts of speech is not common. 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 passive stem (tägäddäl-) and causative stem (asgäddäl-) using prefixes tä- and as-,

[1] For transcription purpose, IPA representation is used with some modification.

respectively. Other verb forms are also derived from roots in a similar fashion.

Verbs are inflected for person, subject, object, gender, number, and tense [2]. Table 2 shows how a perfective Amharic verb inflects for person, subject, gender and number. Other elements like negative markers also inflect verbs in Amharic.

Table 2. Inflection of a perfective verb

Person	Singular	Plural
1st	säbbärku/hu	säbbärn
2nd masculine	säbbärh/k	
2nd feminine	säbbärš	säbbäräčču
2nd polite	säbbäru	
3rd masculine	säbbärä	
3rd feminine	säbbäräčč	säbbäru
3rd polite	säbbäru	

From the above brief description of Amharic word morphology it can be seen that Amharic is a morphologically rich language. It is this feature that makes development of language models for Amharic challenging. The problems posed by Amharic morphology to language modeling were illustrated by [17] who, therefore, recommended the development of sub-word based language models for Amharic.

1.2. Language Modeling

In language modeling, the problem is to predict the next word given the previous words [13]. It is fundamental to many natural language applications such as automatic speech recognition (ASR) and statistical machine translation (SMT). LM has also been applied to question answering, text summarization, paraphrasing and information retrieval [5].

The most widely used language models are statistical language models. They provide an estimate of the probability of a word sequence W for a given task. The probability distribution depends on the available training data and how the context has been defined [10]. [25] indicated that large amounts of training data are required in statistical language modeling so as to ensure statistical significance.

Even if we have a large training corpus, there may be still many possible word sequences which will not be encountered at all, or which appear with a statistically non-significant frequency (data sparseness problem) [25]. In morphologically rich languages, there are even individual words that might not be encountered in the training data irrespective of its size (Out of Vocabulary words problem). Morphologically rich languages have a high vocabulary growth rate which results in high perplexity and a large number of out of vocabulary words [22]. As a solution, sub-word units are used in language modeling to improve the quality of language models and consequently the performance of applications that use the language models ([6]; [24]; [9]; [12]; [8]).

We have developed sub-word (morpheme-based) language models for Amharic. As to our knowledge, this is the first attempt made for this language. Section 2 presents the development of the language models and the perplexity results obtained. But, before that we would like to discuss about the evaluation metrics used in language modeling.

1.3. Evaluation Metrics

The best way of evaluating language models is measuring its effect on the specific application for which it was designed [15]. However this is computationally expensive and hard to measure. An alternative 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 model will assign a higher probability to the test data [11]. Both cross entropy and perplexity are computed on the basis of this probability.

Cross-entropy of a language (sequence of words) W according to a model $m = P(w_i/w_{i-N+1}...w_{i-1})$ can be calculated as:

$$H(W) = -\lim_{N\to\infty} \frac{1}{N} \log P(w_1 w_2 \ldots w_N) \quad (1)$$

Where, N is the number of tokens in a test text. When N is sufficiently large, cross entropy can be calculated based only on our probability model as follows:

$$H(W) \approx -\frac{1}{N} \log P(w_1 w_2 \ldots w_N) \quad (2)$$

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

Perplexity is a related evaluation metric, which is used most commonly and computed as:

$$PP = 2^{H(W)} \quad (3)$$

$$= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \quad (4)$$

$$= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P\left(w_i / w_1 \ldots w_{i-1}\right)}} \qquad (5)$$

Perplexity can be interpreted as the branching factor of a language model. Therefore, models with low perplexity values are better models. As it can be seen from equation 5, a higher conditional probability of the word sequence leads to lower perplexity. Thus, minimizing perplexity is equivalent to maximizing the test set probability [11]. Because perplexity is the most commonly used evaluation metric, we also evaluated our language models on the basis of perplexity values.

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. When we have different token counts, models can only be compared on the basis of the probability they assign to the test sets.

2.Data Preparation

2.1. The Corpus

A text corpus consisting of 48,090 sentences and 1,542,697 tokens has been prepared. The electronic text is obtained from ethiozena archive which contains written newscast. Since the target application domain is speech recognition, the text has been normalized accordingly.

After normalization, the text corpus has been merged with another one prepared by [17] from the same domain. The combined text corpus, used in the experiment, consists of 120,261 sentences or 2,348,151 tokens or 211,178 types. Table 3 presents the frequency distribution of words in the combined text corpus.

Table 3. Word frequency distribution

Frequency	Number of words
1	121329
2 - 10	69538
11 - 100	17358
101 - 1000	2655
1001 - 10000	293
10001 - 20000	3
above 20000	2

As it can be noted from Table 3, more than 50% (121,329) of the words occur only once (hapax legomena) in the corpus. This indicates the morphological richness of the Amharic language. Although much effort has been exerted to clean the data, there are still misspelled words and correcting them is difficult, as there is no available spelling checker for the language. The existence of misspellings may also contribute to the large number of hapaxes. However, our corpus is not the only one to include large number of hapaxes. Zemánek (2005) 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 10000. These words are function words such as wəsəṭ 'in'.

2.2. Morphological Analysis

Developing a sub-word language model requires to have a word parser which splits word forms into its constituents. Different people ([1]; [20]; [16]) have attempted to develop morphological analyzer for Amharic using different methods. However, none of the systems can be directly used for this project. The systems developed by [1] and [20] suffer from lack of data. The morphological analyzer developed by [16] seems to exhibit a dearth of lexicon. It has been tested on 207 words and it analyzed less than 50% (75 words) of the words. Moreover, the output of the system is not directly useful for this 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.

An alternative approach is offered by unsupervised corpus-based methods which do not need annotated data. These methods are particularly interesting for resource scarce languages like Amharic.

Two freely available, language independent unsupervised morphology learning tools have been identified: Linguistica [7] and Morfessor [14]. Both tools have been tried on a subset of our corpus (9996 sentences). Unfortunately, it has been found out that Linguistica divides every word into exactly 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 for the subsequent experiments.

Morfessor requires a list of words as an input. 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. To compare these findings with the situation in Amharic, two word lists have been prepared from the corpus: a list of tokens and a list of types.

Since Morfessor has been trained on two different word lists, there are two outputs (morph segmentation) and, therefore, two kinds of morph-segmented corpora: token_based_corpus and type_based_corpus. Token_based_corpus is a morph corpus where the morphs have been found by analyzing the list of tokens whereas in type_based_corpus the morphs have been found by analyzing the word type list.

3.Experiments

3.1. Morpheme-based Language Models

The tool used for language modeling purpose is SRI Language Modeling toolkit (SRILM) [19]. SRILM is a freely available open source language modeling toolkit.

Each corpus 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 for both corpora. A significant difference in perplexity (860.47 for the token_based_corpus and 117.43 for the type_based_corpus) has been observed. The reason for this difference might be due to 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 [14] that segmentation is less common when word tokens are used as data. Accordingly, only the type_based_corpus has been used for subsequent experimentation.

N-gram models of order 2 to 5 have been tried. 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 4.

Table 4. Perplexity results

N-gram	Smoothing technique	Perplexity
4gram	Good-Turing with Katz backoff	113.24
5gram	Absolute Discounting with 0.7 discounting factor	112.79
5gram	Witten-Bell	110.88
5gram	Natural Discounting	117.37
4gram	Modified Kneser-Ney	107.54
5gram	Unmodified Kneser-Ney	103.63

As it can be seen from Table 4, the best performing model is a 5gram model with unmodified Kneser-Ney smoothing. This result is in line with the finding of [18] 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 tried only for these three smoothing techniques because SRILM toolkit supports interpolation only for them. Table 5 shows the best results for each smoothing technique.

Table 5. Perplexity results with interpolation

N-gram	Smoothing Techniques	Perplexity
4gram	Witten-Bell	112.1
5gram	Modified Kneser-Ney	101.38
4gram	Absolute Discounting with 0.7 discounting factor	118.38

Interpolating n-gram probability estimates at the specified order n with lower order estimates sometimes yield better models [19]. Our experiment verified this fact. A 5gram 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. The best performing model has a perplexity of 102.59 on the evaluation test set.

As indicated by [19], discarding unknown words or treating them as a special "unknown word" token affects the quality of language models. Thus, unknown words[2] have been mapped to a special "unknown word" token for the best model indicated in Table 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.

3.2. Word-based Language Models

To compare these results, 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 6 shows the perplexity of word-based models. The 5gram model with unmodified Kneser-Ney is the best model compared with the other word-based language models.

Table 6. Perplexity of word-based models

N-gram	Smoothing technique	Perplexity
3gram	Good-Turing with Katz backoff	1151.29
5gram	Absolute Discounting with 0.7 discounting factor	1147.04
5gram	Witten-Bell	1236
5gram	Natural Discounting	1204.14
4gram	Modified Kneser-Ney	1107.32
5gram	Unmodified Kneser-Ney	1078.16

Interpolation of n-gram probability estimates has also been tried for the three smoothing techniques for which SRILM

[2]sub-word units are considered as words in sub-word based language models

supports interpolation. As it can be seen from Table 7, improvement with interpolation has been achieved for a 5gram model with modified Kneser-Ney. The other two smoothing techniques have lower perplexity values without interpolation.

Table 7. Perplexity of word-based models with interpolation

N-gram	Smoothing Techniques	Perplexity
5gram	Witten-Bell	1241.41
5gram	Modified Kneser-Ney	1059.38
3gram	Absolute Discounting with 0.7 discounting factor	1158.63

The optimal quality has been obtained with 5gram 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.

The perplexities of our word-based language models are very high compared to what has been reported by [17], where the maximum perplexity of a bi-gram word-based language model was 167.889. To discover the reason behind the difference, we have developed word-based language models using our corpus in the same fashion as [17] did.

In [17] HLStats, HBuild and HSGen modules of the HTK toolkit [25] have been used since the version of the HTK toolkit used did not incorporate HLM language modeling toolkit. HLStats create 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. The perplexity is high compared to the one reported by [17], but this is not a surprise to us since the size of the training corpus used in our experiment is larger.

The problem with this method 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 same experiment is conducted repeatedly, the perplexity values also vary from experiment to experiment, as the sentences generated are different. Therefore, we can not directly compare the perplexity of the word-based language models of our experiment with the one reported by [17] because the test sentences used to calculate the perplexities are completely different.

3.3. Influence of Data Quality

Although we expect that the high perplexity of our word-based language models to be mainly due 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. About 10,000 sentences of our corpus have been manually checked for spelling errors and merged with the data used in [17] for the speech recognition experiments. This forms data_set_I that consists of 21,922 sentences and 425,359 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.

Table 8. Word-based models with data_set_I

N-gram	Smoothing technique	Perplexity
4gram	Absolute Discounting with 0.7 discounting factor	981.464
4gram	Witten-Bell	1091.03
5gram	Natural Discounting	1013.81
3gram	Modified Kneser-Ney	970.285
3gram	Unmodified Kneser-Ney	940.046

Table 9. Word-based models with data_set_II

N-gram	Smoothing technique	Perplexity
4gram	Absolute Discounting with 0.7 discounting factor	988.073
5gram	Witten-Bell	1096.71
4gram	Natural Discounting	1022.22
3gram	Modified Kneser-Ney	986.471
3gram	Unmodified Kneser-Ney	955.999

As it can be observed from Table 8 and 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 10 and 11.

Table 10. Interpolated word-based models data_set_I

N-gram	Smoothing technique	Perplexity
4gram	Absolute Discounting with 0.7 discounting factor	979.125
4gram	Witten-Bell	1084.92
4gram	Modified Kneser-Ney	936.898

Table 11. Interpolated word-based models data_set_II

N-gram	Smoothing technique	Perplexity
4gram	Absolute Discounting with 0.7 discounting factor	987.89
4gram	Witten-Bell	1092.23
4gram	Modified Kneser-Ney	953.953

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) have been observed as a result of mapping unknown words to "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 from spelling errors. This enables us to conclude that correcting spelling errors did not reduce the high perplexity of word-based models and, therefore, the sole source for the high perplexity is the morphological feature of the language.

3.4. Comparison of Sub-word and Word-based models

The perplexity values of word-based and morph-based models are not 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 high probability is considered as a better model. To avoid underflow, log probabilities are considered and, therefore, we actually compared the log probabilities.

The total log probability of the best performing morph-based model (A 5gram model with Kneser-Ney smoothing and interpolation of n-gram probability estimates, indicated in Table 4) is -834495. Whereas, the corresponding word-based model has a total log probability of -705218. Table 12 depicts the log probabilities of best morph-based model and the corresponding word based model which has a perplexity of 1059.38 (see Table 7).

Table 12. Log probabilities I

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

The best performing word-based language model (5gram 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 morph-based model is -836215 although its perplexity is 102.26. Table 13 shows this fact. This tells us that word-based models have high log probability and, therefore, are the better models although their perplexity is higher.

Table 13. Log probabilities II

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

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 morphologically rich languages in general, and Amharic in particular.

4.Conclusion

In this paper we described an attempt to develop sub-word based language models for Amharic. Since Amharic is one of the less resourced languages, we have used freely available softwares or toolkits (Morfessor for morphological parsing and SRILM for language modeling) in the course of our experiment.

Substantial reduction in the out of vocabulary rate, which is a severe problem in morphologically rich languages, has been observed as a result of using sub-words. In this regard, using sub-word units is preferable for the development of language models for Amharic. Low perplexity values have been obtained with morph-based language models. However, when comparing the quality based on the probability assigned to the test sets, word-based models seem better. Therefore, recognition experiments will be necessary to study the utility of the models in a particular application scenario.

We also observed that the output of the morphological analyzer consists of unsegmented words that should have been segmented. Efforts along this line might also improve the morph-based model.

No attempt has been made so far to deal with the non-concatenative root-pattern morphology of the language. A complete morphological decomposition of a semitic language will include affix segmentation as well as decomposition into root and pattern. Thus, a word in Amharic can be decomposed into root, pattern and one or more affix morphemes. Mere consideration of these morphemes as a language modeling unit might result in loss of word level dependencies since the root consonants of the words may stand too far apart. Therefore, new approaches, which capture word level dependencies, for modeling semitic languages in general, and Amharic in particular are

required. Building a separate model for root consonants and the other morphemes (patterns and affixes), and interpolating the models might help to capture word level dependencies. Currently, we are working in this direction.

5.Acknowledgment

We would like to thank University of Hamburg for financial support. Our thanks also goes to Solomon Teferra Abate who allowed us to use his text corpus.

6.References

[1] Abiyot Bayou (2000) Developing Automatic Word Parser for Amharic Verbs and Their Derivation, M.Sc. thesis, Addis Ababa University, Addis Ababa.

[2] Baye Yemam (1986 EC.) yäamarəña säwasäw. Addis Ababa: EMPDE

[3] Bender, M. L., J. D. Bowen, R. L. Cooper and C. A. Ferguson (1976) Language in Ethiopia. London:Oxford University Press.

[4] Ethnologue (2004) Available at: http://www.ethnologue.com/show_language.asp?code=AMH

[5] Gao, Jianfeng and Lin Chin-Yew (2004) "Introduction to the Special Issue on Statistical Language Modeling" ACM transactions On Asian Language Information Processing. 3(2): 87-93

[6] Geutner, P. (1995) Using morphology towards better large-vocabulary speech recognition systems. In Proceedings of ICASSP, 445-448.

[7] Goldsmith, John. (2000) Linguistica: An Automatic Morphological Analyzer. The Proceedings from the Main Session of the Chicago Linguistic Society's Thirty-sixth Meeting. Arika Okrent and John Boyle (eds.) 36-1.

[8] Hirsimäki, Teemu et. al. (2005) Morphologically Motivated Language Models in Speech Recognition. In Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR).

[9] Ircing, P., P. Krebc, J. Hajic, S. Khudanpur, F. Jelinek, J. Psutka and W. Byrne (2001) On large vocabulary Continuous Speech Recognition of Highly Inflectional Language – Czech. In Proceeding of the European Conference on Speech Communication and Technology.

[10] Juqua, Jean-Claude and Jean-Paul Haton (1996) Robustness in Automatic Speech Recognition: Fundamentals and Applications. London: Kluwer Academic Publishers.

[11] Jurafsky, Daniel and James H. Martin (2006) Speech and Language Processing: An Introduction to Speech Recognition, Computational Linguistics and Natural Language Processing. Draft of 2nd edition. Available at: http://www.cs.colorado.edu/~martin/slp2.html

[12] Kirchhoff, Katrin et al. (2002) Novel Speech Recognition Models for Arabic. Johns-Hopkins University Summer Research Workshop, Final Report. Available at: http://ssli.ee.washington.edu/people/katrin/arabic_resources.html

[13] Manning, Christopher D. and Hinrich Schütze (1999) Foundations of Statistical Natural Language Processing. London: The MIT Press.

[14] Mathias Creutz and Krista Lagus (2005) Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora using Morfessor 1.1. Publications in Computer and Information Science, Report A81, Helisinki University of Technology.

[15] Rosenfeld, Ronald (1997) Statistical Language Modeling and N-grams. Available at: http://www.cs.cmu.edu/afs/cs/academic/class/11761-s97/WWW/tex/Ngrams.ps.

[16] Saba Amsalu and Dafydd Gibbon (2005) Finite State Morphology of Amharic. In Proceedings of RANLP, Bulgaria, P. 47 – 51.

[17] Solomon Teferra Abate (2006) Automatic Speech Recognition for Amharic. Ph.D. Thesis Available at: http://www.sub.uni-hamburg.de/opus/volltexte/2006/2981/pdf/thesis.pdf

[18] Stanley F. Chen and Joshua Goodman (1998) An Emperical Study of Smoothing Techniques for Language Modeling. Available at: http://people.csail.mit.edu/regina/6864/slides/goodman.pdf

[19] Stolcke, Andreas (2002) SRILM - An Extensible Language Modeling Toolkit. Available at: http://www.speech.sri.com/projects/srilm/

[20] Tesfaye Bayu (2002) Automatic Morphological Analyzer for Amharic: An Experiment Employing Unsupervised Learning and Autosegmental Analysis Approaches. M.Sc. Thesis, Addis Ababa University, Addis Ababa.

[21] Titov , E. G. (1976) The Modern Amharic Language. Moscow:Nauka Publishing House.

[22] Vergyri, Dimitra, Katrin Kirchhoff, Kevin Duh and Andreas Stolcke (2004) Morphology-Based Language Modeling for Arabic Speech Recognition. Available at: http://www.speech.sri.com/cgi-bin/run-distill?papers/icslp2004-arabic-lm.ps.gz.

[23] Voigt, R. M. (1987) "The classification of central Semitic" Journal of Semitic Studies, 32: 1 21.

[24] Whittaker, E. W. D. and P. C. Woodland (2000) Particle-based language modeling. In Proceeding of International Conference on Spoken Language Processing, Beijing, China.

[25] Young, Steve, Dan Kershaw, Julian Odell, Dave Ollason, Valtcho Valtchev and Phil Woodland (2000) The HTK Book. Available at: http://nesl.ee.ucla.edu/projects/ibadge/docs/ASR/htk/htkbook.pdf.