Design and Development of Enhanced Morphological Analyzer for Ge’ez Verbs Using Memory Based Learning Algorithms

Gebremeskel Hagos Gebremedhin
Masters Candidate, College of Software Engineering, Nankai University, China
Frank Chao Wang
Associate Professor, College of Software Engineering, Nankai University, China

Abstract:
This paper is carefully designed for Ge’ez morphological analyzer. Ge’ez is the classical language of Ethiopia and still used as the liturgical language of Ethiopian Orthodox Tewahedo church. Many ancient literatures were written in Ge’ez. The literature includes religious texts and secular writings. The ancient philosophy, tradition, history and knowledge of Ethiopia were being written in Ge’ez. Morphological analyzer is one of the most important basic tools in automatic processing of any human language and analyses the naturally occurring word forms in a sentence and identifies the root word and its features.

In this paper, MBL is used to automatically analyze the morphology of Ge’ez verbs via the concept of machine learning for training and analysis. TiMB’s IB2 and TRIBL2 algorithms have been used for the implementation. The performance of the system has been evaluated using 10-fold cross validation technique on the default and optimized parameter settings. The overall accuracy with optimized parameters using IB2 and TRIBL2 was 94.24% and 93.31%, respectively. Similarly, the overall precision, recall and F-score with optimized parameters using IB2 were 55.6%, 56.3% and 59.95%, respectively. In the same manner the precision, recall and F-score using TRIBL2 were 58.8%, 60.3% and 59.54%, respectively. Moreover, a learning curve was drawn. The graph showed that as the number of training dataset increase, the accuracy on unseen data can be increased. Therefore, IB2 algorithm shows better result than TRIBL2 algorithm for Ge’ez verb morphology.

Keywords: Ge’ez Morphology, Ge’ez verbs, morphological analyzer, memory based learning, character based analysis, cross validation, feature extraction

1. Introduction
Language is one of the fundamental aspects of human behavior which constitutes crucial components of our lives. It is also the best mechanism through which information and knowledge can be kept for a long period and passed on from one generation to another. In its written form, it serves as a means of recording information and knowledge on a long term basis and transmitting what it records from one generation to the next. In its spoken form, it serves as a means of coordinating our day-to-day life with others (Allen, 1995).

Linguistics can be defined as the study of languages, particularly, natural languages. Natural language is a set of conventions used by human beings for communication purposes (Akmajian, 1995). The academic discipline that studies the computer processing of natural language (NL) is known as Natural Language Processing (NLP). Morphology is the field within linguistics that studies the identification, analysis and description of the structure of a given language’s morphemes and other linguistic units, such as root words, affixes, parts of speech [3,4]. Morphology attempts to formulate rules that model the knowledge of the speakers of those languages.

Natural Language Processing (NLP) has been developed in the 1960s, as a sub field of Artificial Intelligence and Linguistics [5]. It is a field of Computer Science that investigates interactions between computers and human languages, which is used for both generating human readable information from computer systems and converting human language into more formal structures that a computer can understand [6]. Therefore, it is important for scientific, economic, social, and cultural reasons. It is experiencing rapid growth as its theories and methods are deployed in a variety of new language technologies. For this reason, it is important for a wide range of people to have a working knowledge of NLP. Within the industry, this includes people in human computer interaction, business information analysis, and web software development. Within academia, it includes people in areas from humanities computing and corpus linguistics through to computer science and artificial intelligence [7].

NLP originally referred to as Natural Language Understanding (NLU) in the early days of artificial intelligence. It is well agreed today that while the goal of NLP is true NLU, that goal has not yet been accomplished. A full NLU System would be able to: paraphrase an input text, translate the text into another language, answer questions about the contents of the text and draw inferences from the text. While NLP has made serious inroads into accomplishing goals one to three, the fact that NLP systems cannot, of themselves, draw inferences from text, NLU still remains the goal of NLP [8]. The aim of
NLP is studying problems in the automatic generation and understanding of natural languages. Natural language is understood as a tool that people use to express themselves and has specific properties that reduce the efficiency of textual information retrieval systems. These properties are linguistic variation and ambiguity. NLP studies the problems of automated generation and understanding of natural human languages [9].

While NLP is a relatively recent area of research and application, as compared to other information technology approaches, there have been sufficient successes to date that suggest that NLP-based information access technologies will continue to be a major area of research and development in information systems now and far into the future [8]. Current NLP systems tend to implement modules to accomplish mainly the lower levels of processing. This is for several reasons. First, the application may not require interpretation at the higher levels. Secondly, the lower levels have been more thoroughly researched and implemented. Thirdly, the lower levels deal with smaller units of analysis, e.g., morphemes, words, and sentences, which are rule-governed, versus the higher levels of language processing which deal with texts and world knowledge, and which are only regularity-governed [8]. Well known problems of NLP are morphological analysis, part of speech tagging, word sense disambiguation, and machine translation [6].

Morphology deals with the inner structure of individual words and the laws concerning the formation of new words from pieces, morphs [10]. Morphologies are motivated by four Considerations: (1) the discovery of regularities and redundancies in the lexicon of a language (such as the pattern in walk, walks, walking; jump, jumps, jumping); (2) the need to make explicit the relationship between grammatical features (such as nominal number or verbal tense) and the affixes whose function it is to express these features; (3) the need to predict the occurrences of words not found in a training corpus; and (4) the usefulness of breaking words into parts in order to achieve better models for statistical translation, information retrieval, and other tasks that are sensitive to the meaning of text [11].

Furthermore, Morphology is the field within linguistics that studies the [12] identification, analysis and description of the structure of a given language’s morphemes and other linguistic units, such as root words, affixes, parts of speech, etc. [8, 9]. Morphemes are the smaller elements of which words are built. Two broad classes of morphemes are stems and affixes. Affixes that are added to the base to denote relations of words are morphemes. Morphemes can either be free (they can stand alone, i.e., they can be words in their own right, are also referred as roots), e.g., dog, or they can be bound (they must occur as part of a word), e.g., the plural suffix –s on dogs [12].

Morphology plays two central roles in language. In its first role, derivational morphology (morphemes) allows existing words to be used as the base for forming new words with different meanings and different functionality. Example, the noun judgment is formed from the verb judge, and the adjective inedible has a different semantic meaning from its related verb eat. In its second role, inflectional morphology (morphemes) deals with syntactic features of the language such as person (I am, you are, he is), number (one child, two children), gender (actor, actress), tense (eat, eats, eating, eaten, ate), case (he, him, his), and degree (cold, colder, coldest). These syntactic features, required to varying degrees by different languages, do not change the part of speech of the word (as the verb eat becomes the adjective inedible) and do not change the underlying meaning of the word (as cellist from cello) [14].

Morphologically complex languages are challenging due to the combinatorial explosion of possible morpheme structures [15]. Ge’ez is one of the morphologically very complex languages [4] and study the verb morphology is the base for other word categories. The advent of personal computers and mobiles has increased communication between speakers using written texts of Ge’ez language. Many electronic documents are produced due to the worldwide communication. Those produced electronic documents need several automatic natural language processing systems such as information retrieval, machine translation, grammar checking and the like to reduce the size of a word list to a manageable level and improve retrieval performance, and capture the strong relationships existing between different word forms in the language. Mutual understanding makes day to day life easy by human-computer interaction to solve problems through NLP applications. The morphology of a language is base for all other NLP applications. Artificial intelligence is the very interesting research area for implementing human activities simply via machines effectively and reliably. Memory based learning algorithms are the key concepts for making machines intelligent to perform tasks and processes. The process to create mutual understanding among the users in terms of time and cost besides a reliable result of data analysis and quality is mostly expected area of research for providing intelligent system that can identify the linguistic nature of natural languages. This is the interesting work for the use of languages by any user in the world by giving clear clue of that language since the morphology is a basic to do using the model of NLP and machine learning though providing the ability of automatically learn and improve from experiences without being explicit program i.e. class of algorithms solve the problem of users easily and automatically to solve the problem.

Machine learning is used for the development of intelligent system which are helpful in day today life. Morphological analyzer is one of the most important basic tools in automatic processing of any human language. It analyses the naturally occurring word forms in a sentence and identifies the root word and its features. In spite of its significance, some languages do not have any morphological analyzers available. The absence of such a tool for research severely impedes the development of language technologies and applications like natural language interfaces, machine translation, etc. In these languages [12]. Traditionally, morphological analysis has been done manually. It took linguistic experts several months to years in order to describe a single language. More recently, computer algorithms have been applied which can automate and therefore speed-up the process. These algorithms deploy techniques from machine learning to improve their performance with increasing numbers of words or analyzed word examples they have access. Morphology and its analysis play an important role in many natural language processing tasks. Automatic speech recognition (ASR) is concerned with the identification of spoken words and their transformation into text.
Morphologically complex languages are especially challenging due to the combinatorial explosion of possible morpheme structures [14].

Morphological analyzer will also be an important component of the NLP to be developed for Ge'ez language. Thus, the beneficiaries of this research include researchers who need to take part in achieving the goal of developing efficient NLP system for Ge'ez language. The research paper will be used to put linguistically motivated structure of Ge'ez verbs, to help Ge'ez learners and to develop higher forms of NLP systems such as automatic dictionary (lexicon) compilation, spell-check, machine translation, speech recognition, POS tagging, automatic sentence construction, morphological synthesizer, and etc of NLP researches applications.

Ge'ez is the classical language of Ethiopia and belongs to the Semitic language family. The other Semitic-Ethiopian languages are Tigre and Tigrinya (known as North Ethiopian), Amharic (the national language of Ethiopia), Argobba, Harari, and Gurage (called South Ethiopian). Although Ge'ez ceased to be spoken in the twelfth or thirteenth century, it has remained the language of literature and of liturgy. Knowledge of the language derives from the vast literature written in Ge'ez. The literature includes religious texts (such as the Bible, Apocrypha, Pseudepigrapha, liturgical literature, homiletic, theological, and magical texts, stories of martyrs and saints, religious poetry, hymns in honor of Christ, the Virgin, the martyrs, the saints, and angels), as well as secular writings (histories and romances, legal, mathematical, and medical texts) [16]. The ancient philosophy, tradition, history and knowledge of Ethiopia was being written in. Ge'ez and also there are different books which are written in this language. To keep and transfer these identities to the next generation, citizens must know the meaning of these written books/documents. If they do not know the idea in the documents, they will not give any attention for these heritages. These resources can also be used as sources of philosophy, creativity, knowledge and civilization both to Ethiopia and the rest of the world. Moreover, if someone who wants conduct a research on issues related to the classical custom, history, politics, tradition, and religion of Ethiopia, he/she has to explore the works handed down from the previous generations. So, he/she must investigate these literatures. To use these resources, one must know the language itself or else these literatures have to be translated into either of the currently spoken languages manually, which may take a long time. To solve this problem and the previous one, studying the linguistic nature of the language in a scientific approach with the help of Information Technology is necessary. This requires conducting a research to develop a more efficient morphological analyzer for Ge'ez verbs to minimize the time to study the language and to extract automatically the documents which were written in Ge'ez language.

2. Deign of Morphological Analyzer for Ge'ez Verbs

Memory based language processing is based on the idea that learning and processing/performance are two sides of the same coin [17]. The learning component is memory based as it involves storing examples in memory without abstraction, selection, or restructuring. In the performance component of a MBLP system the stored examples are used as a basis for mapping input to output; input instances are classified by assigning them an output label. During classification, a previously unseen test instance is presented to the system. The class of this instance is determined on the basis of an extrapolation from the most similar example(s) in memory. There are different ways in which this approach can be operationalized [18].

Memory-based learning (MBL) is an approach to NLP based on a symbolic machine learning method. Memory-based learning is based on the assumptions that in learning a cognitive task from experience people do not extract rules or other abstract representations from their experience, but reuse their memory of that experience directly. Memory based learning and problem solving incorporates two principles [19]: learning is the simple storage of a representation of experiences in memory, and solving a new problem is achieved by reusing solutions from similar previously solved problems. This simple idea has appeared in many variations in works in artificial intelligence, psychology, statistical pattern recognition, and linguistics [17]. Generally, the central claim of the MBL paradigm is that decisions about new facts are based on re-use of stored past experiences. Therefore, the main goal of an MBL model is to extrapolate the class of new exemplars based on their similarity to stored exemplars. As described in [17, 18, 21], memory-based learning has a great future potential to analyzing Natural language processing (NLP) tasks from spelling error correction to machine translation and automatic extraction of knowledge from text.

All experiments with memory-based learning described here were carried out using the Tilburg Memory-based Learner, or TiMBL. This software package is developed and maintained by the Induction of linguistic knowledge group at Tilburg University, and is a very efficient feature rich implementation. It can be run either as a ‘one-shot’ command line classifier or as a server, and there is also a C++ API that makes it possible to integrate the memory-based learning functionality into other programs such as python [18]. With this module, all functionality is exposed through the C++ interface that is also available to Python scripts. Being able to access the API from Python greatly facilitates prototyping TiMBL-based applications [20].

2.1. System Architecture Design

As described in [2, 17] the task of performing a full morphological analysis of a word form is usually considered as a segmentation of the word into morphemes, combined with an analysis of the interaction of these morphemes. Segmentation of morphemes determines the syntactic class of the word form as a whole. Morphological analysis of a language is a non-trivial task even for languages with highly inflected. The memory-based learning approach of morphological analysis primarily concerns saving or learning of some patterns of the morpheme in memory and trying to
classifying and analyzing the newly or unseen words by analogy [22]. In this section we designed and described morphological analyzer of Ge’ez verbs.

![Figure 1: Architecture of the Proposed Geez Verbs Morphological Analyzer](image)

### 2.2. Phases of the Analyzer

The morphological analyzer has two phases, as shown in Figure 1. The first phase is a training phase which consists of morpheme annotation to manually annotate inflected Ge’ez verbs, and feature extraction to create instances in a fixed length of windows and the memory-based learning to train the dataset. On the other hand, the morphological analysis phase contains the feature extraction (instance making) to deconstruct a given text, morpheme identification to classify and extrapolate, stem and root extraction to label segmented inflected words with their morpheme functions. Morpheme annotation was done manually since Ge’ez verb morphemes are mostly expressed by internal phonological changes in the root. Because of this the internal irregular changes of phonemes make the morphological analysis more complex. As shown in Figure 1, during preparing annotated dataset for experimentation purpose the following tasks were identified and performed in the order listed.

- Identifying inflected words
- Segmenting the word into prefix, stem, suffix
- Putting boundary marker between each
- Describing the representation of each marker

NegPref/PosPref/PreCirc + Stem + SufCirc/SMS/OMS

After the annotated verbs are stored in a database, features are extracted automatically from the manually created morphological database to make instances using an Algorithm 4.1 based on the concept of windowing method in a fixed length of left and right context which is the average word length in the database. As described in [2], windowing method is dividing the windows where the instances are placed in the left and right context to hold fixed-length string of features, which describe the linguistic context of the token to be classified. Each instance is associated with a class. The class represents the morphological category in which the given word possesses. Instances consisting of a fixed number of features are created by windowing methods-by sliding each character down as a focus letter for each character in the left and right context. Each example focuses on one letter, and includes a fixed number of left and right neighbor letters, in this case using a 10–1-10-1 window which yields twenty-two features. The input character in focus, plus the ten preceding and ten following characters are placed in the windows. As stated in [1, 18] the complex morphological analysis is placed at the right most part as a class.

The MBL (TiMBL) is founded in the hypothesis that the extrapolation of behavior from stored representations of earlier experience to new situations, based on the similarity of the old and the new situation. The TiMBL algorithms IB1, IB2, IGTREE, TRIBL and TRIBL2 are used to develop the knowledge model.
As described in [24], concept descriptions are determined by how the IB2 algorithms selected similarity and classification functions use the current set of saved instances by calculating Similarity Function: This computes the similarity between a training instance and the instances in the concept description. Similarities are numeric-valued.

Classification Function: This receives the similarity function’s results and the classification performance records of the instances in the concept description. It yields a classification for each instance and a Concept Description Updater: This maintains records on classification performance and decides which instances to include in the concept description. Inputs include i, the similarity results, the classification results, and a current concept description. It yields the modified concept description.

The second, morphological analysis phase, includes instance making to make the input words to be suitable for memory based learning classification, the morpheme identification to classify and extrapolate the class of new instances, the stem extraction to reconstruct and insert identified morphemes, and finally the root extraction to get root forms of verb stem with their grammatical functions for the implementation of the training phase as shown in Figure 1 through calculating the measures.

\[ \delta(v_1, v_2) = \sum_{i=1}^{n} |P(C_i|v_1) - P(C_i|v_2)| \]  
(Eq 1)

3. Experiment and Results

The experimental testing was done using TiMBL’s default and optimized parameter settings. Based on the default values of parameters of each algorithm (IB2 and TRIB2), we obtained results shown in Tables 1 and 2 in each training and testing experiments (10-fold cross validation).

| No of Experiment | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| No instances     | 11029 | 10739 | 10629 | 11017 | 10727 | 11042 | 10935 | 10979 | 10963 | 11155 |
| Seconds Taken    | 0.211 | 0.193 | 0.175 | 0.116 | 0.157 | 0.159 | 0.118 | 0.114 | 0.244 | 0.279 |
| Size of Instance | 186980 | 1889840 | 1790880 | 1862720 | 1973680 | 1936880 | 1951320 | 1933120 | 1922840 | 1850960 |
| Base (byte)      | 56.65 | 55.98 | 57.30 | 56.73 | 58.28 | 56.29 | 56.07 | 56.07 | 56.20 | 58.18 |
| Compression      | 94.9  | 95.7  | 93.5  | 94.4  | 99.4  | 96.5  | 92.1  | 77.3  | 84.5  |       |
| Accuracy         |       |       |       |       |       |       |       |       |       |       |

Table 1: 10-Fold CV Experiment on IB2 with Default Setting and b=5330

In Table 1, b=5330 is the number of instances, to be taken from the top of the training file, to act as the bootstrap (sample with replacement) set of memorized instances before IB2 starts adding new instances. It is only applicable in
conjunction with IB2 algorithm. The value (5330) here is the average value of b from the 10-fold experiments which scores high performance in each fold.

| No of Experiment | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------------|---|---|---|---|---|---|---|---|---|----|
| No Instances     | 11029 | 10739 | 10629 | 11017 | 10721 | 10442 | 10935 | 10979 | 10943 | 11155 |
| Seconds Taken    | 0.0738 | 0.0565 | 0.060 | 0.046 | 0.089 | 0.0707 | 0.0765 | 0.0473 | 0.0814 | 0.117 |
| Size of Instance | 3743280 | 3705960 | 3757200 | 3756240 | 3836560 | 3769840 | 3804200 | 3833800 | 3802960 | 3709760 |
| Base (Byte)      | 56.07 | 55.82 | 55.50 | 55.96 | 55.20 | 56.21 | 55.83 | 55.65 | 56.95 | 57.09 |
| Compression      | 96.1 | 97.9 | 96.8 | 96 | 93.1 | 93.9 | 97.9 | 93.9 | 75.9 | 70.5 |

Table 2: 10-Fold CV Experiment on TRIB2 with Default Setting

The overall performance of the experiment is the average value of each fold experiment results. The average experiment result of the two algorithms with default parameter and optimized parameter setting described in Table 3 and Table 6 respectively.

| Evaluation Method | Algorithm | Compression (%) | Time Taken (Seconds) | Size of Instances Base (Byte) | Accuracy (%) |
|-------------------|-----------|-----------------|----------------------|-------------------------------|--------------|
| 10-fold CV        | IB2       | 56.48           | 0.177                | 1898204                       | 91.72        |
|                    | TRIBL2    | 55.98           | 0.0725               | 3761076                       | 91.19        |

Table 3: Average Performance of 10-fold CV Experiment with Default Parameter Setting

As we observe from Table 3 IB2 algorithm perform a little bit better generalization of the accuracy of the model than TRIBL2 algorithm. Moreover, IB2 algorithm used much less memory storage than TRIBL2. But IB2 consumes much more time than TRIBL2 to process.

Many classifiers are parameterized and their parameters can be tuned to achieve the best result with a particular dataset. In most cases it is easy to learn the proper value for a parameter from the available data [23]. After tuning a number of parameters along with the algorithms, we identified the parameters with high performance as optimized parameters. These are nearest neighbor: k=5, distance metric: modified value difference metric (-mM), feature weighting: information gain (-w2) and distance-weighted class voting: inverse distance (-d ID). We did the experiments with those parameter settings and we have got the results shown in Tables 4 and 5.
As shown in Tables 3 and 6 the generalization performance of IB2 algorithm with default and optimized parameter setting is 91.72% and 94.24% respectively. Similarly, the generalization performance of TRIBL2 algorithm with default setting is 91.19% and with optimized parameters is 93.31%. Therefore, IB2 shows better general performance than TRIBL2 in both default and optimized parameter settings. It also performs better in compression the instance trees and uses less memory than TRIBL2. On the other hand, TRIBL2 processes within short seconds than IB2 on the same number of instances.
Aside from the percentage of correctly classified test instances, we used some more evaluation metrics that have become common in information retrieval and machine learning namely precision, recall, and F-score (f-measure). Precision can be defined as the percentage of correct morph boundaries among all morph boundaries suggested by the system and recall is the percentage of correct boundaries discovered by the classifier. F-score or f-measure is a harmonic mean of precision and recall. It is a commonly used metric to summarize precision and recall in one measure [10]. The precision, recall and F-score for unknown words with default and optimized parameters described in Table 7.

| Algorithm | With Default Parameters | With Optimized Parameters |
|-----------|-------------------------|---------------------------|
|           | Precision | Recall | F-score | Precision | Recall | F-score |
| IB2       | 52.9      | 52.1   | 52.49   | 55.6      | 56.3   | 59.95   |
| TRIBL2    | 55.4      | 56.6   | 55.99   | 58.8      | 60.3   | 59.54   |

Table 7: 10 folds CV Results of Average Precision, Recall and F-score with Default and Optimized Parameter Settings

As shown in Table 7 optimizing some of the parameters achieved better result than default parameter settings on both algorithms. This result is achieved with small dataset (1105 - which does not include all verb categories complete inflection). We observed that the ill-balanced distribution of dataset matters the extrapolation of new instances. On the other hand, it has negative impact on the general performance of the system. Our dataset contains complete derivation of one verb and some sample verbs from the remaining categories. Thus, we believe that if our dataset contains more inflected verbs which include complete inflected verbs of each category, the system will achieve better result than this. So, it is possible to say that, the result obtained here is acceptable to implement the system in large scale examples and words with more complexity.

In memory-based learning the minimum size of the training set to begin with is not yet specified. However, as stated in many literatures, the size of the training data matters the learning performance of the algorithm. Hence, it is crucial to draw learning curves in addition to reporting the experimental results [17]. We perform series of experiments on systematically increased amounts of training material up to the currently available total dataset which is 1105. In most cases to draw a learning curve, the learning can be measured by fixing the test set against which the increased model is systematically tested. We took the test set by calculating the average of 10% of each number of training set and set here to be 70.

| No of Instances | Training Words | Performance (%) |
|-----------------|----------------|-----------------|
|                 | 49             | 69.9            |
|                 | 885            | 78.5            |
|                 | 1199           | 81.2            |
|                 | 1949           | 82.9            |
|                 | 3325           | 86.8            |
|                 |                | 86.7            |
|                 |                | 86.1            |
|                 |                | 86.4            |
|                 |                | 86.4            |
|                 |                | 86.1            |
|                 |                | 86.4            |
|                 |                | 86.4            |
|                 |                | 86.4            |
|                 | 1116           | 94.5            |
|                 | 11924          | 99.8            |

Table 8: Generalization Accuracy with Increasing Number of Words

The learning curve displays on the Ge'ez verbs morphology test set, with increasing amounts of instances in the training set (subsets which was labeled in the X-axis were simply created by taking the n first instances of the full training set). To include all the dataset, we start simply taking 5 and 90 as initial training set and the rest taking multiple of 90. The x-axis has a logarithmic scale. The curve shows that at the maximal amount of training material currently available (1080 words in the 10% split training set) the curve has not flattened; with larger amounts of training material better generalization accuracies on unseen data can be expected. The curve suggests that if the training set is doubled in size about three or four times more, a 100% score on the test set appears to be in reach. In general, we can infer from the learning curve that more training data gives better generalization.

Figure 3: Learning Curve with Increasing Number of Words
3. Discussion

The proposed morphological analysis experiment was held with 10-fold cross validation. The system is trained on approximately 90% of the corpus and then tested on the remaining 10%. The performance of the system in terms of accuracy, classification time and memory usage were determined by evaluating the morpheme identification by training the system with the default and optimized algorithmic parameter settings. The evaluation criteria used for morpheme identification were accuracy, recall, precision, and F-score. The accuracy of the model with default settings are 91.72% and 91.19% for IB2 and TRIBL2 classifiers respectively. After a number of experimentations of one parameter against the other; the nearest neighbor is 5, the distance metrics is MVDM, the feature weighting metric is IG and the class voting weight is inverse-distance, taken as optimized parameters for both IB2 and TRIBL2 results better performance. Therefore, using those optimized parameters the generalization accuracy of IB2 and TRIBL2 becomes 94.24% and 93.31% respectively.

IB2 algorithm showed significant performance even though it takes more time than TRIBL2. Like Amharic morphology, we believe that the problem of speed may not be a serious problem for Ge'ez morphological analysis because the fundamental concern is obtaining a morphological analyzer which performs better in terms of accuracy. Therefore, IB2 classifier is better than TRIBL2 regardless of more processing time.

The precision, recall and F-score were also calculated by taking the average of the 10-fold cross validation. The results using IB2 algorithm with default parameter settings are 52.9%, 52.1% and 52.49 % respectively. Similarly, TRIBL2 classifier was also evaluated in the same manner with IB2 and obtained 55.4%, 56.6%, 55.99% precision, recall and F-score, respectively. These algorithms are also evaluated using optimized parameters to obtain good result than the default ones. Therefore, we obtained 55.6%, 56.3% and 59.95% precision, recall and F-score respectively using IB2 algorithm. In the same manner we obtained 58.8%, 60.3% and 59.54% precision, recall and F-score respectively using TRIBL2 algorithm. In general, both algorithms have insignificant difference. Therefore, they are suitable for Ge'ez morphological analysis.

As described in in the introduction section, to the best knowledge of the researcher, Ge'ez language morphological analyzer has been investigated by Desta Berihu. The study limited to ቦሮም /kata/) (he killed) category verb forms. The author has analyzed the morphology of Ge'ez verbs using rule-based approaches specifically CV-based and Two-Level Morphology (TLM) to design the model and to implement the prototype of the analyzer. The experimental result the author found an accuracy of 92.05% at feature level and of 73.98% at verb level.

This paper develops a model and test machine learning (MBL) perspective of Ge'ez morphological analysis which includes all category verb forms. During the experiment we used 10-fold cross-validation technique. The experimental result shows that the generalization performance of IB2 and TRIBL2 algorithms are 94.24% and 93.31% respectively. Comparison of morphological analysis of similar natural languages which developed by using memory-based learning approach shown in Table 9.

| Morphological Analysis of | Evaluation Techniques | Size of Dataset (in words) | Word Class                  | Algorithm | Accuracy (%) |
|---------------------------|-----------------------|----------------------------|-----------------------------|-----------|--------------|
| Dutch language            | 10-fold CV            | 247,415                    | Verbs, Nouns, Adjectives    | IB2       | 94           |
| Arabic language           | 10-fold CV            | 16,626                     | Verbs, Nouns, Adjectives    | IB1       | 15           |
| Swedish language          | 10-fold CV            | 4,189                      | Verbs, Nouns, Adjectives    | IB1       | 85.6         |
| Swahili language          | WER                   | 9,700                      | Verbs, Nouns, Adjectives    | IB1       | 13.3 (86.7)  |
| Amharic language          | 10-fold CV            | 1022                       | Verbs, Nouns, Adjectives    | IB1       | 93.59        |
|                           |                       |                            |                             | IGTREE    | 82.26        |
| Ge'ez language (This work)| 10-fold CV            | 1105                       | Verbs                       | IB2       | 94.24        |
|                           |                       |                            |                             | TRIBL2    | 93.31        |

Table 9: Comparison of Ge'ez Morphological Analysis with related works

As shown in Table 9, except Swahili morphological analysis, the evaluation techniques are similar. Swahili morphological analysis was evaluated using word error-rate (WER) technique. The lower the WER means the system will be better. Ge’ez is a complex inflected language; it is difficult to address and analyze all the morphological features. Due to this, this paper addressed the morphological analysis of Ge’ez verbs only. Verbs are morphologically the most complex POS in Ge’ez. The experimental result shows that the generalization performance of IB2 and TRIBL2 using optimized parameters 94.24% and 93.31%, respectively. This result is achieved with small dataset which does not include all verb categories complete inflection. We observed that the ill balanced distribution of dataset has negative impact on the general performance of the system. The dataset contains complete deriviation of one verb and some sample verbs from the remaining categories. This due to time limitation to collected all complete inflected verbs. Thus, we believe that if our dataset contains more inflected verbs which include complete inflected verbs of each category, the system will achieve better result than this. So, it is possible to say that, like other languages, the result obtained here is acceptable to implement the system in large scale examples and words with more complexity.
4. Conclusion

In this paper the memory-based learning for Ge’ez verbs is achievable. Based on the experiment results obtained in the previous chapter, memory-based learning showed a good result for morphological analysis of Ge’ez verbs relative to small number of datasets. The unavailability of complete inflection verbs of all Ge’ez categories verbs and annotated morphological database forced us to spend much more time in preparing dataset. This is also the main reason for the small number of our datasets. We annotated manually 1105 verbs to be suitable to TiMBL algorithms. From these annotated verbs, we extracted 12135 instances automatically. This data set was divided into training and testing data from which 90% for training and 10% for testing. The training data is used to assess how much the model is able to learn and the test (unseen) data used for evaluating the performance of the algorithms. By adjusting the default and optimized parameter settings of TiMBL tools, we trained and tested our dataset. To do this we used both IB2 and TRIBL2 algorithms.

We found that IB2 is good at memory usage on both default and optimized settings (with 91.72% and 94.24% accuracy) but it has low processing speed which in turn takes more time. On the other hand, TRIBL2 algorithm performs a little bit different from IB2. It performs 91.19% and 93.31% with default and optimized parameter settings respectively. TRIBL2 classifier needs more memory usage and high-speed during training and test of dataset. Therefore, there is tradeoff between both algorithms is respective of their advantages. In summary, memory storage and speed may have a matter in choosing from both algorithms for Ge’ez morphological analysis.

Future works based on shortcomings of the current study are outlined here. In other words, the tasks which were not included in our work are described as a recommendation for future works to make a complete full flagged morphological analysis.

- This paper addressed the morphological analysis of Ge’ez verbs and with relatively very small datasets. This can be extended to nouns, adjectives, compound words and complex verbs. The use of a large training data has huge importance in enhancing the performance of the system. In other words, initiation of a big project to develop an efficient full-fledged automatic morphological analyzer for Ge’ez language which works for all POS categories is required.

- The complex nature of Ge’ez like Amharic morphology the system cannot segment some words. So, managing spelling changes (inserting, deleting) and extracting roots are possible project areas.

- A comparative analysis for Ge’ez and other local languages, using other approaches like HMM, SVM, MBL, ILP and finite state morphological analysis can be done.

5. References

i. J. Allen, Natural language understanding, CA, USA: Benjamin-Cummings Publishing Co. Inc, 1995.
ii. Antal van den Bosch and Walter Daelemans, Memory based morphological analysis for Dutch language: computational linguistics, Tilburg University, 1999.
iii. Bender Myron Lienel, Language in Ethiopia, Oxford University Press, London, pages 23-27 and 99-106, 1976.
iv. Desta Berihu, Design and Implementation of Automatic Morphological Analyzer for Ge’ez Verbs using rule-based approach: Athesis submitted to the school of graduate studies of the Addis Ababa University in partial fulfillment for the degree of masters of Science in computer science, 2010.

v. Saranya Kittanakom, Morphological Analyzer for Malayalam Verbs, a project report submitted in partial fulfillment for the award of the degree of Master of Technology in 95 Computational Engineering and Networking, Amrita Vishwa Vidyapeetham Amrita School of Engineering, Coimbatore, 641105, 2008.

vi. KorayAlk, Olay Taner Yildiz, Unsupervised Morphological Analysis Using Tries: Department of Computer Science and Engineering. Isik University, 2011.

vii. Steven Bird, Ewan Klein and Edward Loper, Natural Language Processing with Python, 1sted. USA, 2009.

viii. Edwared Liddy, Natural Language Processing: In Encyclopedia of Library and Information Science, 2nd Ed. NY.
ix. Marcel Decker, Inc, 2001
x. Jisha Jayan and Rajeev Rajendran, Morphological Analyzer and Morphological Generator for Malayalam - Tamil Machine Translation, Tamil University, Thanjavur, 2011.
x. Walter Daelemans, J. Zavrel, A. Bosch, et al, TiMBL: Tilburg Memory-Based Learner version 6.3 Reference Guide: ILK Technical Report – ILK10-01, 2010, http://ilk.uvt.nl/timbl/http://ilk.uvt.nl/timbl/
x. John Goldsmith, Segmentation and morphology: university of Chicago, 2009.

xii. Anand Kumar, Morphology Based Prototype Statistical Machine Translation System for English to Tamil Language, A Thesis Submitted for the Degree of Doctor of Philosophy in the School of Engineering, Amrita School of Engineering Amrita Vishwa Vidyapeetham Coimbatore-641 112, Taminadu, India, 2013.
xiii. Richard Wicentowski, Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework: A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy; Baltimore, Maryland, 2002.

xiv. Sebastian Reiner Spiegel, Machine learning for the Analysis of Morphologically Complex languages: a thesis submitted to the University of Bristol in accordance with the 93 requirements for the degree of Doctor of Philosophy, Merchant Venturers School of Engineering University of Bristol, 2011.
xv. Mark Aronoff and Kirsten-Fudeman, What is Morphology, 2nd Edition, Blackwell Publishing, 2011.
xvi. Wolf Leslau, Comparative Dictionary of Ge’ez (Classical Ethiopic): Ge’ez English/ English-Ge’ez with an index of the Semitic roots, Wiesbaden Harrassowitz, ISBN- 1-447-02592-1, (1987, 1991).
xvii. Walter Daelemans and Antal van den Bosch, Memory-Based Language Processing, University of Antwerp and Tilburg University, 2005.

xviii. Walter Daelemans and Antal van den Bosch, Memory-based language processing, CLiPS Research Group, University of Antwerp, ILK / Tilburg centre for Creative Computing, Tilburg University, The Netherlands, 2009.

xix. Walter Daelemans, Introduction to the special issue on memory-based language processing. ILK, Computational Linguistics, Tilburg University, the Netherlands, pp 287-296, 1999

xx. Sander Canisius and Maarten van Gompel, Python timbl, Tilburg University, 2006, [Available] on http://github.com/proycon/python-timbl/

xxi. Anders Nklestad, A Machine Learning Approach to Anaphora Resolution Including Named Entity Recognition, PP Attachment Disambiguation, and Animacy Detection, 2009.

xxii. Mesfin Abate, Amharic Morphological Analysis Using Memory Based Learning, A Thesis Submitted in Partial Fulfillments of The Requirement for the Degree of Master of Science in Computer Science, Addis Ababa University, Ethiopia, 2014.

xxiii. Payam Refaelzadeh, Leitang Huanliu, Cross validation, Arizona State University, 2008.

xxiv. Daha Aha, JohnKibler and JolmAlbert, Instance based Algorithms in Machine learning, Volume. 6, pp. 37-66, 1991.