Rule-Based Morphological Analyzer for Amharic Verbs

Chalachew Seyoum
HiLCoE, Computer Science Programme, Ethiopia
drrakke@gmail.com

Wondowssen Mulugeta
HiLCoE, Ethiopia
School of Information Science, Addis Ababa University, Ethiopia
wondwossen.mulugeta@aau.edu.et

Abstract

Although Amharic is spoken by more than 20 million people, little research has been done in developing effective and efficient Natural Language Processing (NLP) applications for the language. This paper presents the design and implementation of a rule based morphological analyzer for Amharic verbs that is accurate, fast and easy-to-use. Consonant-Vowel (CV) template and two-level morphology based approaches were used in designing the model for the morphological analyzer. Morphological rules of the language were extracted from literature on Amharic grammar and represented in the design and prototype using regular expressions. The prototype, which was developed using Python programming language, also used lexicons consisting of root verbs and the CV-templates they inflect with to convey different grammatical information. Once the analyzer was designed and the prototype developed, it was tested using 1500 Amharic verbs extracted from a web crawler. The analyzer was able to finish the analysis in approximately 50 seconds and achieved an accuracy of 85.6%, thereby satisfying the initial goal of the research.

Keywords: Rule-based; Morphological Analyzer; Amharic Verbs; Consonant-Vowel Template; Two-level Morphology; Lexicon

1. Introduction

Morphological analysis is an essential component of any natural language processing system. It helps to find the minimal units of a word which holds linguistic information for further processing [1]. Learning and understanding the minimal units of a word in any language can help us develop language processing tools for that language. In morphologically complex languages, morphological analysis is also a core component in information retrieval, text summarization, question answering, machine translation, etc. [1].

There are currently two mainstream methodologies for developing a morphological analyzer: rule-based and corpus-based [1]. The first approach uses hand-coded suffix replacement rules and lexicon for stemming while in the second approach, rules are derived from a corpus automatically. The first approach being language specific requires considerable linguistic expertise to craft rules, but it can result in higher performance. In the second approach, rules are derived from a corpus automatically [2].

Corpus-based morphological analyzers are easier to construct than rule-based morphological analyzers. But their performance level is highly related to the degree to which the corpus used to train the model is developed. In languages where there are well enriched corpuses, performance of corpus-based morphological analyzers can match and even outperform rule-based morphological analyzers. However, not every language has a well-developed corpus. Amharic is one of such languages.

Amharic, the working language of the Federal Democratic Republic of Ethiopia and some of its regional governments, is a Semitic language and, according to the 2007 national census, is spoken by more than 21 million native speakers and is ranked as the second most spoken language, next to Arabic, in the Semitic family and 50th world wide [3]. Amharic by nature is a rich language. This richness is more than partially owed to the fact that the language has a complex morphology. This has made learning the language and analyzing its morphology difficult.

Amharic verbs, like other Semitic languages, are very complex [4], consisting of a stem and up to four prefixes and four suffixes. The stem in turn is

composed of a root, representing the purely lexical component of the verb, and a template, consisting of slots for the root segments and for the vowels (and sometimes consonants) that are inserted around and between these segments. The template represents tense, aspect, mood, and one of a small set of derivational categories: passive-reflexive, transitive, causative, iterative, reciprocal, and causative reciprocal [4]. This template and rule of the basic morphology of verbs can be translated into a model for a morphological analyzer.

Although Amharic is spoken by a large portion of Ethiopia's population, there is little progress made in developing computational resources and applications for the language. This is in no small way related to the fact that there has been little research done on the language. Morphological analyzers are an essential component of many NLP applications and as such should be in the forefront of any research efforts to develop effective and efficient NLP applications for Amharic.

Recently, there have been researches done on building rule-based morphological analyzer and generator on Semitic languages like Arabic and Amharic. One such work is HornMorpho, which is a morphological analyzer/generator for three Ethiopian languages. HornMorpho uses a finite state model to carry out word analysis and generation for these languages [5]. Although this model analyzes Amharic verbs with good precision, it has some usability issues. For example, the model sacrifices speed for accuracy which makes analysis rather slow, which makes the integration of the system with other high level NLP applications inefficient. It also outputs results in Roman letters and does the analysis process after transliteration of the input, which could also factor into the speed of the analysis. Aside from these, the analyzer does not output the stem of a verb and can also be a little daunting to use. This is because it does not have a graphical user interface for interaction with the user and relies on the user writing calls to analyzer functions in a Python shell. Therefore, it still can be useful to try and tackle the issue using different methods from the ones already explored and try to come up with a more usable morphological analyzer for Amharic verbs. As essential as morphological analyzers are to NLP applications, it is vital to identify different methodologies to develop them if other high level NLP applications are to be developed for the Amharic language.

Therefore, it is essential to design and develop a morphological analyzer for Amharic verbs which overcomes the shortcomings of the existing model and provides a different approach than the one already used. To do so, the following research questions were raised and answered in this paper.

1. What are the morphological rules that define the morphological properties of Amharic verbs?
2. What are the components of a rule-based morphological analyzer and what approaches to rule-based morphological analysis do we need to follow in order to design a rule-based morphological analyzer for Amharic verbs that can accurately represent their morphological properties?
3. Based on the selected approaches and identified components, how can we design and develop a rule-based morphological analyzer for Amharic verbs using regular expressions that has comparable performance accuracy to existing rule-based morphological analyzers for the language?

Based on answers to these questions, a rule-based morphological analyzer for Amharic verbs with comparable accuracy, faster execution time and easier and better usability was designed and developed in this paper.

2. Literature Review

2.1 Amharic Verb Morphology

Amharic verbs are derived from root consonants, which carry the basic lexical meaning, by intercalating vowel patterns and/or using different derivational morphemes [6]. These roots are mostly comprised of consonants and when intercalated with a CV-pattern, produce a stem with a specific grammatical property which can further be inflected or derived to form different variation of that verb. For example, the root ግድል */gdl/* (kill) can be intercalated with the vowel ኧ */ä/* based on the

perfective form CV-template '$C_1VC_2C_2VC_3$' to form the perfective stem ገደል- */gäddäl-/*.

Amharic verbs unlike nouns and adjectives, which are derived from words with different part-of-speech classification, are mainly derived from verb roots. These verb roots undergo morphological transformation to form stems which are bound and need to be inflected for person, gender and number to find their natural and usable form [7].

Once an Amharic verb goes through the derivation process to form the stem word, the next phase it undergoes is to be inflected for person, gender, number, tense, aspect and mood to find their natural and usable form. This is achieved by affixation of different morphemes with different grammatical purposes to the stem. These morphemes can be prefixes, suffixes, infixes and circumfixes.

To analyze the morphology of any language, it is essential to understand the underlying morphological components that make up the words. However, due to its non-concatenative nature and other morphological processes, the surface form of an Amharic verb seldom matches the collective surface form of its constituent morphemes. Since the surface form is the input used for morphological analysis, it is essential to understand the morphological processes that end up shaping it. These processes are vowel changes, assimilation and alternation rules.

Vowel changes occur when two morphemes are joined together and at least one of them is a vowel. In Amharic verbs this can occur when an end of a morpheme is a consonant or a vowel and the beginning of the next morpheme is a vowel, a different vowel in the case of a vowel to vowel succession. These changes can manifest themselves on or in the stem and the affixes attached. For example, when the prefix አንድ /*?nd*/ is attached to the verb ይሰብር */y-säbbär/*, the final word will have the surface form of አንዲሰብር /*?ndisäbbär*/ where the ድ /*d*/ grapheme of the prefix is changed to ዲ /*di*/ and the ይ /*y*/ grapheme of the verb is eliminated.

Assimilations occur when two morphemes come together and the beginning grapheme of the second morpheme is dropped due to the phonological properties of the last and first graphemes of the first and second morphemes respectively. In Amharic verbs this phenomenon can be seen when the negation marker አል /*all*/ is prefixed to the 2nd and 3rd person negative mood and the ል /*ll*/ morpheme of the negation marker is assimilated to the person marker. For example, the surface form of the negation mood form verb አል-ት-ሰበር */all-t-sbär/* is actually አት-ሰበር */at-sbär/*.

Alternations in Amharic morphology mainly occur due to the interaction between a suffix and the character preceding it. Alternations can be caused when the 1st person marker of the gerundive form, ኤ */e/*, or the feminine marker morpheme ኢ */i/* come after certain consonants. These consonants then change to different ones due to the alternation process. Table 1 shows the consonants that undergo the alternation process.

Table 1: Alternation of Consonants

Consonant	*Changed form for 'ኤ'*	*Changed form for 'ኢ'*
ል	ሌ	ሊ
ድ	ጄ	ጂ
ስ	ሼ	ሺ
ት	ቼ	ቺ
ጥ	ጬ	ጪ
ን	ኜ	ኚ
ዝ	ዤ	ዢ

2.2 Components of a Rule-based Morphological Analyzer

In the rule-based approach, two components can usually be distinguished in an analyzer: a declarative component corresponding to linguistic knowledge and a procedural component which represents the analysis strategy. Linguistic knowledge includes the grammar and the lexicon of the language while analysis strategy is an algorithm which specifies in detail each of the operations involved in the process of analysis [8]. We have re-classified these components for the purpose of this research. One component will be a set of rules and algorithms and the other will be a lexicon.

2.3 Set of Rules and Algorithms

The main component of a rule-based morphological analyzer is the set of rules and algorithms that determine the analysis of the input. The rules are built from the morphological rules

governing the formation of words in that language, morphotactics. Also there are rules that manage phonological and orthographical changes which are likely to occur during the concatenation process of the morphemes to produce the inflected forms. Such rules too should be the basic elements of the set of rules formulated to be used by the analyzer for its robust analysis process [3]. The algorithms on the other hand are the computational processes that are responsible for analyzing input words as per the rules discussed above.

2.4 Lexicon

Another component the morphological analyzer needs is a lexicon that contains roots and stems of a word. This will enable the analyzer to effectively analyze inputs into their corresponding morphological features and finally compare the output. The lexicon will also enable the addition of certain constraints and rules as to how these input words can be analyzed correctly. This will make the rule building process less complicated in removing these constraints from the possible elements of the rule set.

3. Related Work

Although many researches on the subject matter exist, the ones that will be discussed in this section are the ones with the most relevance to the subject matter.

HornMorpho is a set of Python programs for analyzing and generating words in Amharic, Tigrinya, and Oromiffa built by Gasser [5]. It is a morphological analyzer/generator that, while analyzing, takes input words, analyzes them and returns a result comprising of the root word, the component morphemes and the different grammatical features they represent and while generating takes a stem and morphemes and generates a verb using those.

It uses Finite State Transducers (FST) to model the different morphological rules of the generator/analyzer. FSTs are a more complex version of Finite State Machines (FSM) which are computational models consisting of a finite number of states and no memory such that a certain state of such a machine only depends on the preceding state [9, 10, 11]. However, FSTs not only accept or reject strings of one language but also transform them into strings of another language [9].

A morphological analyzer is developed by Desta Berihu that takes an input Ge'ez word and analyzes it into its constituent morphemes and the stem [3]. The author used rule-based approach, specifically CV-based and Two-Level Morphology (TLM), to design the model and to implement the prototype of the analyzer. Besides, the analyzer uses a knowledgebase as a demon while identifying the morphosyntactic features. Algorithms that take into consideration the morphological, morpho-phonological and orthographic properties of Ge'ez language are also the main components of the analyzer.

An analyzer for Arabic was built by Shaalan and is modeled using an Augmented Transition Network (ATN) to represent the context-sensitive knowledge about the relation between a stem and its inflectional additions [8]. The ATN consists of arcs, each of which is a rule that links a departure node to a destination node, called states. More than one rule may be associated with one arc which allows actions, such as omit affix or convert radical letter, to be associated with each arc. The algorithm starts from the first node of the ATN and non-deterministically generates analyses of the input when it reaches the final node. In this model different set of rules will apply on the word segments and features of the lexical entry of the root to determine the morpho-syntactic features of the input inflected word.

4. Design and Implementation of the Amharic Verb Morphological Analyzer

4.1 Architecture of the Morphological Analyzer

In designing the morphological analyzer, CV-based and two-level morphology based approaches were used. The CV-based approach was used to distinguish between different stems of a given verb. CV-templates of verbs, which they use to derive different stems from root verbs, were used to classify verb stems into different verb forms such as imperfective, perfective or gerundive. The CV-templates were made a part of the root verb lexicon which will be discussed in more detail later. Two-level morphology based approach was then used to

deconstruct an input into its component morphemes, which means changing its surface form into the lexical equivalent. Since Amharic is a non-concatenative language and changes its lexical form due to factors like assimilation, alternation and vowel changes, this process is essential if we are to find every morpheme that makes up the input verb and correctly analyze it.

The morphological analyzer is designed to take inputs in their natural Ge'ez orthography, analyze them and output their lexemes/stems, the different morphemes that are attached to them and their corresponding morphosyntactic features. The analyzer parses, identifies and strips the different morphemes attached to the stem of the word and identifies the morphosyntactic features using rules that are represented as regular expressions in 'if-else' conditions within the model. Aside from these rules, the analyzer uses lexicons to identify the root of the verb and simplify the parsing process.

During analysis, the analyzer takes an input, an Amharic verb, and compares it to a lexicon to identify the root verb and the CV-template it is inflected with. Once the root is identified, the analyzer then converts the input into a format that makes analysis easier. The analyzer then takes the formatted input and proceeds to detaching the prefixes and suffixes attached to the input and while doing so identifies and assigns the different grammatical features this affixes represent. Finally, the analyzer outputs the root verb, stem and the grammatical features identified during the analysis.

4.2 The Prototype

The prototype was implemented using Python programming language and was designed for ease of use. The user will only have to copy and paste the verb or verbs to be analyzed once the prototype starts running and click the 'Start Analysis' button to start the analysis. The prototype then outputs the results of the analysis as shown in Figure 1.

Once the analysis is completed, the analyzer then generates an output consisting of the root verb, stem, verb form, prefixes, suffixes, subject, object and object function. The example in Table 2 shows how the analyzer takes an input and outputs its analysis result.

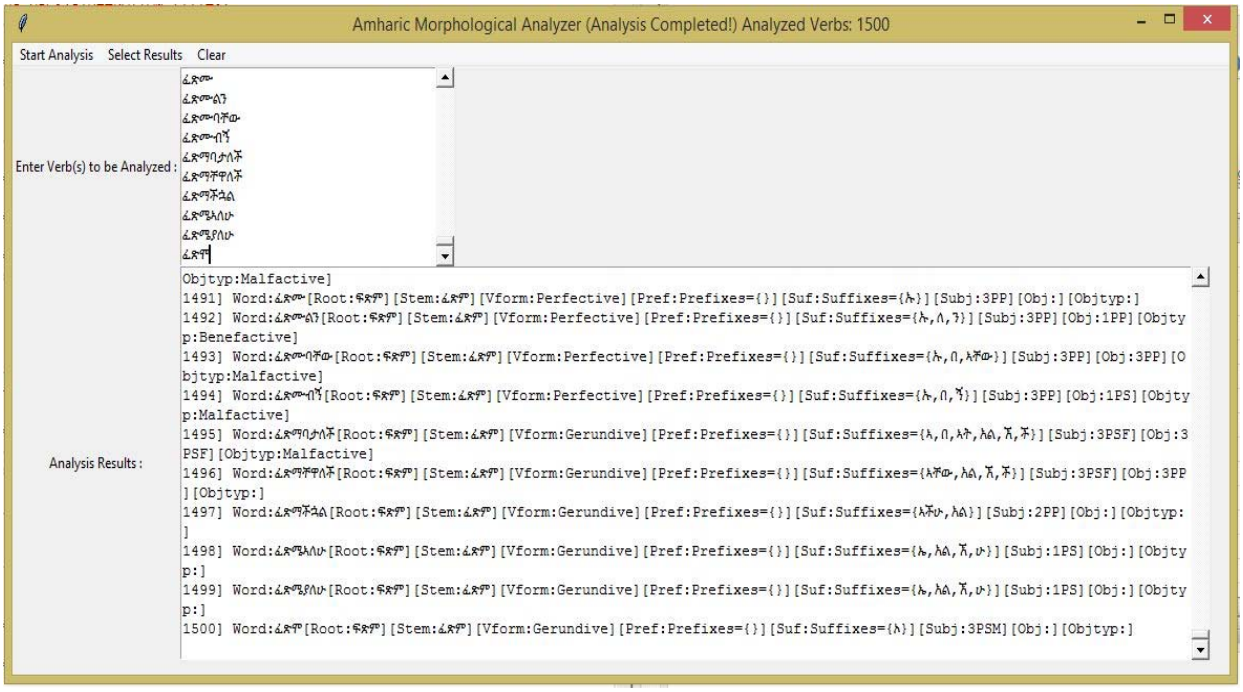

Figure 1: Prototype of the Amharic Verb Morphological Analyzer

Table 2: Sample Analysis Result from the Morphological Analyzer

Word: አስብሬያለሁ [Root: ስብር][Stem: አስብር][Vform: Gerundive][Pref: Prefixes={አስ}][Suf: Suffixes={ኤ,አል,ኧ,ሁ}][Subj: 1PS][Obj:][Objtyp:]

5. Testing and Evaluation

To test the prototype, 1500 Amharic verbs were chosen for their morphological complexity. The analyzer was then tested multiple times using the same test data to calculate the average time it took to finish analysis. During every test, the analyzer was able to finish the analysis of the 1500 Amharic verbs in approximately 50 seconds.

Once the test was concluded, accuracy was calculated overall and at grammatical feature level. To calculate the overall accuracy, the total number of correctly analyzed verbs was divided by the number of verbs in the test set and multiplied by 100.

Table 3 shows the calculated overall accuracy of the prototype morphological analyzer.

Table 3: Overall Accuracy of the Prototype Morphological Analyzer

Result Category	*Total No. of Results*	*Percentage*
Correctly Analyzed Verbs	1284	85.6%
Incorrectly Analyzed Verbs	216	14.4%
Total	*1500*	*100%*

Accordingly, of the 1500 verbs tested, 1284 (85.6%) were correctly analyzed while 216 (14.4%) were incorrectly analyzed. This makes the overall precision of the prototype 85.6% where all grammatical properties of the analysis result were correct.

To calculate the accuracy at grammatical feature level, the total number of correctly analyzed grammatical features, based on similarity, was divided by the total number of that grammatical feature in the analysis result and multiplied by 100.

Table 4 shows the calculated accuracy for each grammatical feature. As can be seen from Table 4, the prototype has an accuracy of above 95% for most grammatical features except for identifying verb forms. The accuracy of this feature is almost equal to that of the overall accuracy which indicates that the overall performance is undeniably tied to how well the prototype analyzes verb form correctly. Therefore, future work on improving the accuracy of the analyzer when identifying verb form will undoubtedly increase the overall performance of the morphological analyzer.

Comparison of Amharic Verb Morphological Analyzer with HornMorpho 2.5

As discussed earlier, HornMorpho is a morphological analyzer/generator for Amharic, Tigrigna and Oromiffa. The morphological analyzer we built was compared to HornMorpho to understand how well it performs against an existing morphological analyzer. Performance of the morphological analyzers was compared based on their overall accuracy and the speed with which they were able to analyze the given inputs. Aside from performance, size of the prototypes on disk and the ease with which users were able to use them were also used as metrics for the comparison.

Table 4: Accuracy of the Prototype Feature Level

Result Category	*Correctly Analyzed Features*	*Incorrectly Analyzed Features*	*Total*	Accuracy in Percent
Root	1448	52	1500	96.53 %
Stem	1425	75	1500	95 %
Verb Form (Grammar)	1298	202	1500	86.53 %
Prefix	1463	37	1500	97.5 %
Suffix	1478	22	1500	98.5 %
Subject	1433	67	1500	95.5 %
Object	1463	37	1500	97.5 %
Object Function	1500	0	1500	100 %

To calculate performance, 200 verbs were randomly selected from the 1500 verbs in the original test set and given to the morphological analyzers. The speed with which the analyzers were able to

finish analysis was then recorded and overall accuracy calculated from the analysis results. Table 5 shows the comparison results of the two morphological analyzers.

Table 5: Comparison between HornMorpho 2.5 and the Amharic Verb Morphological Analyzer Prototype

Comparison Metrics	*HornMorpho 2.5*	*Amharic Verb Morphological Analyzer*
Overall Accuracy	195 Verbs (97.5%)	164 Verbs (82%)
Analysis Speed	3 Mins 20 Secs	8 Secs
Size on Disk	1.24 GB	173 KB
Ease of Use	Function calls on Python Shell	GUI

As can be seen from Table 5, HornMorpho has a better analysis accuracy with 97.5% of the tested verbs being correctly analyzed while our morphological analyzer yielded an accuracy of 82%. On the other hand, our morphological analyzer was able to finish analyzing the 200 verbs in approximately eight seconds while HornMorpho took more than three minutes to finish analysis. When comparing size on disk, the HornMorpho prototype has a size of 1.24 GB while our morphological analyzer has a size of 173 KB. Finally, when it comes to ease of use, our morphological analyzer, because of its graphical user interface, is easier to use than HornMorpho which requires users to write calls to functions in a Python shell.

In conclusion, as can be seen from the analysis results, although HornMorpho has a better accuracy, our morphological analyzer exhibited better speed, smaller size on disk and better ease of use than its counterpart.

6. Conclusion and Future Works

The purpose of this paper was to design and implement a fast, accurate and easy to use rule-based Amharic verb morphological analyzer capable of identifying the different morphological properties of a given input verb. During the process of designing the morphological analyzer, different sources were reviewed to give a better understanding of how to accomplish the task. Literature on the formation of Amharic verbs and their morphological properties was analyzed heavily to understand the rules governing the different components that make up an Amharic verb. Also literature on different techniques for developing a morphological analyzer was reviewed to understand the state-of-the-art of morphological analyzers.

Based on the result of the test performed, this work showed that a fast, easy to use, and accurate rule-based morphological analyzer could be designed and developed using regular expressions and Python programming.

Our morphological analyzer should by no means be the end product. In fact it should serve as a basis for other NLP applications like grammar checkers and text translation applications. For example, our analyzer with the help of a part-of-speech tagger can be used to identify and analyze different verbs found in a text document and this analysis process can enable future research to develop grammar checker for such text documents.

Morphological synthesis is also another future work. Since the rules for the formation of Amharic verbs are what are used in reverse to analyze a given input, synthesis will entail doing the said analysis process backwards. Any future work can find this work helpful in providing the rules for the formation of Amharic verbs.

References

[1] Mesfin Abate, “Development of Amharic Morphological Analyzer Using Memory-Based Learning”, Unpublished masters Thesis, Addis Ababa University, Ethiopia, 2014.

[2] Utkarsh Kapadia and Apurva Desai, “Rule Based Gujarati Morphological Analyzer”, Department of Computer Science, Veer Narmad South Gujarat University, India, 2017.

[3] Desta Berihu Weldegiorgis, “Design And Implementation of Automatic Morphological Analyzer For Ge’ez Verbs”, Unpublished Masters Thesis, Addis Ababa University, November 2010.

[4] Michael Gasser , “A Dependency Grammar for Amharic”, School of Informatics and Computing Indiana University, Bloomington, USA.

[5] Michael Gasser, “HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya”, School of Informatics and Computing, Indiana University, Bloomington, USA, 2011.

[6] Martha Yifiru Tachbelie, “Morphology-Based Language Modeling for Amharic”, University of Hamburg, Hamburg, Germany, August 2010.

[7] Baye Yimam, “Yeamaregna Sewasew”, 3rd Edition, Addis Ababa University Printing Press, 2017.

[8] Khaled Shaalan, “Rule-based Approach in Arabic Natural Language Processing”, International Journal on Information and Communication Technologies, Vol. 3, No. 3, June 2010.

[9] Sebastian Reiner Spiegler, “Machine Learning for the Analysis of Morphologically Complex Languages”, PhD Dissertation, University of Bristol, England, April 2011.

[10] J. Hopcroft and J. Ullman, “Introduction to Automata Theory, Languages, and Computation”, Addison-Wesley, 1979.

[11] M. Sipser, “Introduction to the Theory of Computation", Second Edition, February 2005.