DESIGNING A RULE BASED STEMMER FOR GE'EZ TEXT

[1]Zigju Demissie Baye
[1] Department of Computer Science and Engineering,
[1]Punjabi University, Patiala, India

Abstract: Stemming could be a pre-processing stage for text mining application and often used for Natural Language Processing (NLP). The aim of developing stemmers is used to minimize the inflectional forms and rarely derivationally associated kinds of a word to a typical root form. Ge'ez language is morphologically complicated. This is often because of several words can be formed by means of the varied concatenations of affixes. For the testing, I used affix removal techniques and for evaluation of the stemmer, manually error counting technique was used. From the experiment, three types of errors are detected: over stemmed, under-stemmed, and structural faults.

Index Terms - **Morphology, Stemming, Conflation.**

I. INTRODUCTION

Morphology describes how various forms of words are created, and studies structures of words in the language. Suffixing and Prefixing are the main and common ways of creating word variations in natural language text. Morphology (the internal structure of words) can be broken down into two subclasses: inflectional and derivational. Inflectional morphology describes predictable changes of words which have no effect on a word's part of speech, and these are because of syntax, such as changes in person, number, tense and gender. The second type morphology is derivational which may affect a word's meaning in part of speech. For example, affix changes from adjective to nouns, from verb to nouns, from noun to verb, and so on; like friend, friendly, friendliness, and friendship.

Stemming is an important task in natural language programming (NLP), linguistic morphology, machine translation (MT) and information retrieval (IR). Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conflation. As an example, "computes", "Computing", "computation", "computational", and "computationally" are common variances of a stemmed word "compute".

- **Ge'ez Language**

Ge'ez is that the ancient written communication of Ethiopia and is that the language was operated by both the country state and Orthodox Church starting from the beginning time of Christianity to Ethiopia through the Aksumite era (4th century AD) forward. Over the last two millennia, Ethiopian monastics, scholars and historians have written and developed several Ge'ez literatures that are essential to any study of Ethiopian history. Therefore, though Amharic replaced Ge'ez for state functions within the eighteenth century, Ge'ez remains a crucial language for studies related with Ethiopian history.

- **Conflation Techniques**

To apply stemming operation, we should conflate a word to its several variations. The conflation of terms or Stemming can be implemented in two ways namely, the manual techniques that uses the regular expression and the automatic techniques. Automatic methods are grouped into four categories namely, affix removal, successor variety, table lookup and n-gram approaches. The affix removal approach can also be more divided into longest match and simple removal techniques [9].

- **Stemmer Classification**

Basically, stemming algorithms can be classified into rule based, statistical, dictionary based and hybrid approaches. Each type has its own ways to find for stem. Rule based stemmer (e.g. Affix removal technique, Porter stemmer, Lovin's stemmer) encodes language specific rules, whereas the other, statistical approach used to find distributions of root elements in a database and to learn the morphology of words from a large corpus of a given language. The stemming algorithm or the stemming algorithm which I used to stem Ge'ez text is the rule-based using affix removal technique.

- **Affix Removal Algorithm**

The affix removal algorithms eliminate prefix, infix or suffix from words to reduce word into common base. Most of the stemmers used these affixes for term conflation. These algorithms depend on two principles one is iteration, which removes strings in each

order class one at a time, starting at the end of a word and going towards its beginning. Not more than one matches are allowed in a single order class. The suffix is added to a word in any random order, that is, there exist order classes of suffix. The longest match method is the second way of affix removal in which within any given class of endings, if more than one ending gives a match then longest match should be eliminated [15].

These types of stemmers are also called as **rule-based stemmers**. It is one of the first-born and simplest techniques used in designing of stemmers. These types of algorithms use list of suffixes, prefixes, infixes and/or circumfixes with each affix having the measures under which it can be eliminated from a word to get a valid stem or the root. The affix removals based on rules are either done based on longest match basis or in iterative manner. [11]

II. RELATED WORKS

Stemming is a famous research problem. Some of the stemmers developed using a specific stemming technieque for different languages are reviewed below.

Morphological property, conflation, affixation, stop words of Ge'ez is discussed in detail on the books and sites written in Amharic language.[12], and [15].

Morphological analysis of Ge'ez verbs using memory-based learning was designed. The overall accuracy with optimized parameters using instance based learning 2 and tree instance based learning was 93.24% and 92.31%, respectively. [7]

A stemmer for Ge'ez text using rule-based approach was studied. For the experiment, two techniques, affix removal and morphological analysis were used. It is mainly focused on derivational and inflectional word variants of the language. It was performed with an accuracy of 82.42%. [1]

The stemmer for Amharic text was developed using a context-sensitive iterative approach that removes both prefixes and suffixes. In the work, the achievement was an accuracy of 95.9% from 1221 sample tasted words of data.[6]

Stemming Tigrinya words for information retrieval was developed with a hybrid approach which is the combination of rule-based approach and dictionary-based approach. The stemmer was measured with two sets of Tigrinya words. The outcomes show that it accomplished an average accuracy of 89.3%.[4]

a stemmer for Punjabi language was designed and developed using a brute force approach which uses a lookup table method. Finally, the average accuracy of our stemmer is 80.73%.[10]

Automatic stemming for Amharic text was designed with an experiment using successor variety approach, with the peak and plateau, entropy and complete word methods and having the accuracy level of 71.8%, 63.95% and 57.99%, respectively.[2]

The declaration of Ge'ez verbs according to the three traditional schools of */qnie/*. The nature and declaration of Ge'ez verbs consistent with the Ethiopian scholars in diverse schools is shown It is helpful for the more clarification on the way of classification of the verbs in the traditional schools of Ethiopia as the analysis was pure which is stated from the linguistics perspective.[14]

Conflation techniques was also discussed and are grouped into manual and automatic techniques.[9]

Different stemming algorithms were also discussed including rule based, statistical, dictionary based and hybrid approaches. [13] and [14].

III. DESCRIPTION OF SOFTWARE SYSTEMS

In Ge'ez fonts, except Unicode supported ones, characters that have diacritic or diacritic markings need quite one computer memory unit in their representation to use one or additional computer memory units (depending on the alphabet written) for the fundamental character and extra one byte for the diacritic marking. For instance, "□" requires three bytes to represent internally, one computer memory unit for "t", another computer memory unit for "." and another one computer memory unit for "e", therefore it is written as "□" = "t.e". This makes hassle to use Ge'ez alphabet/ fidel that has diacritic marks by treating it as one unit. Therefore, the text was initial reworked in to Unicode representation so translated it into American Standard Code for Information Interchange (ASCII) letters. There are three subsystems in the development of rule-based affix removal Ge'ez stemmer. These subsystems are: document pre-processor, rule-based stemmer, and post processor subsystem.

- ✓ **Document pre-processing subsystem**

This subsystem is the primary task to be accomplished. It is the arrangement of the text for more processing and it helps to make fast and accurate processing of the text [2]. This subsystem includes tokenization and stopword removal operations. Lexical analysis or tokenization is the process of converting an input stream of characters into a stream of words or tokens. The tokenizer accepts the input text file, tokenizes it, and writes the tokens into a file. Tokenization is the first stage of automatic indexing, and of query processing. A token or word is defined as a string of characters separated by white space and/or punctuation marks.

- ✓ **Normalization Subsystem**

This subsystem is used to maps similar characters into one common character. In Ge'ez writing system, there are different symbols having the same sound. For example, 'ፅያል', '□□ል', and '□□ል' have the same meaning. The change of the alphabet into one common representation does not cause a meaning difference, but it increases getting the same term with different characters. Even

though this would violate the linguistic rules and the norms of the Ge'ez language in general, it will increase the success rate of getting similar or related words in an online indexing and searching process.

- ✓ **Stemmer subsystem**

This subsystem is applied by checking length of each word iteratively for affix removal process. Lists of stop words, suffixes, and prefixes are prepared and each word in the sample text are checked if there are affixes or stop words in each iteration. If there is the chance of gaining stop words after affix removal, the stop words removal can be applied.

Prefix removal is a process which reads a word and checks the presence of a real prefix, and eliminates it when the condition is satisfied. Some of prefixes which can be attached to the beginning of Ge'ez words are: አ 'a', □'ei', □ 'e', □ 'me',□ 'mu', ዘበ 'zebe'□ 'te',□ 'ti', አን 'an', ወ 'we', አስተ 'aste', □ 'ta', □ 't', □ 'aaa', □ 'ye', □'ya', □ 'y', □ 'nu', □ 'na', □ 'n', □ 'we', □ 'ze' and □'le'. For example, if we take a root word ውዳሴ 'wdasie', when we add a prefix መ 'me' it gives □□ድስ/'meweds'/. Here the prefix መ 'me' is attach to the left side or the beginning of the word.

Suffix removal like prefix removal is a process which reads a word and checks the presence of a real postfix or suffix, and then it eliminates the alphabets appended to the stem of the word depending on the rules designed when the condition is satisfied. Some of list of suffixes are: □ 'n', □ 't', □ 'w', ሰ 'se', ሙ 'mu', □ 'y', □ 'u', □ 'a', □ 'm', ኣን 'an', □ን 'kn', □ 'n', □ 'hu', □ 'ha', □□ 'at', '□ □' ot, □□ 'am', □ 'kä', □ 'ku', ኪ 'ki', □ 'we', ዎ 'wo', □□ 'yat', □ 'ya', □ 'wi', □ 'na', □ 'yä', ከ 'ke' □ 'nä', □□ 'kn', □ □ 'yan', □□ 'kmu', □□ 'wat', □□ 'wya', □□ 'homu', □ ' 'a', □□ 'kemu'. These suffixes can be combined to produce another single suffix. For example, the two suffixes ኩ 'ku' and ክሙ 'kmu' can be appended as a postfix ኩክሙ in a root wors ቀደሰ 'qedese' and gives an inflected result ቀደስኩክሙ 'qedeskukmu'. After postfix removal, the stem became ቀደስ 'qedes'. The postfix checker checks whether the final result of the stemmed word is in stop words list or not. Then if it exists there, it removes, unless it produces the stem of the word.

The sample texts which are used for testing the stemmer were organized from different sources, such as: ውዳሴ ማርያም 'Wdasie Maryam' (Praise to St. Mary, prayer book), ቅዳሴ ሃዋርያት (Kidasie Hawaryat 'Praise of Apostles to God').

Name of text	Total words	Distinct words	Word-ratios in percent	% of words with frequency 1	%words with frequency more than 1
ቅዳሴ ሃዋርያት /Qdasie hawaryat/	819	476	58.11	60.46	39.54
ውዳሴ ማርያም /Wdasie Maryam/	474	299	63.08	64.54	35.46

Table 1: Number of words and their distributions to Ge'ez sample data sets

- **Testing and Evaluation of the Stemmer**

The stemmer is evaluated for its accurateness on the sample input text having 1293 words for analysis. From these 1293 words the 982 words were evaluated as valid and 311 words were invalid because they violated both the exceptional and general rules.

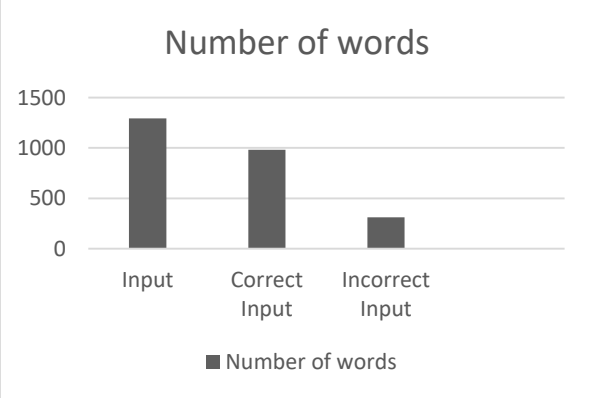

Fig. 1: Graphical representation of evaluation

This system gave 75.95% accuracy. I used the following formula to calculate the accuracy. Accuracy of the stemmed result can be measured using the following equation.

Accuracy (%) = 100 * (Correctly stemmed words / Total words)

Accuracy (%) = 100 * (982/1293)
= 75.95 %

Test data attributes	Total count
Sample text total words	1293
Correct stemmed words	982
Incorrect output words	311
Stop words in the sample text	95

Table 2: Summary for tested data

The problem that causes testing outcome of the system to low performance is that, the domain diversity of the text corpus affected the number of existence of the segment leading to taking inappropriate segment. In addition, the very complex nature of Ge'ez language is the infixes. It is very challenging to develop a rule for infixes since it has very diverse morphological property for different words. So, it makes a complexity to produce the correct stem.

IV. CONCLUSION

The correctness of the stemmer highly affects the results of the system in which it is used. Ge'ez is a language with root pattern structure typical of Semitic languages. A single word in the language has several alternatives. The accuracy of the system can be more improved by growing the Ge'ez words and affixes dictionary. The rule base can be upgraded linguistically by designing rules which can cover every morphological property of Ge'ez language. This can improve the accuracy of the stemmer and enables to hold very large and complex text corpus.

REFERENCES

[1] Abebe Belay Adege (2010), Designing a Stemmer for Ge'ez Text Using Rule Based Approach, Addis Ababa University

[2] Genet Mezemir Fikremariam (2009). Automatic Stemming for Amharic Text: an Experiment Using Successor Variety Approach, Addis Ababa University

[3] Prince Rana (2010). Design and Development of a Stemmer for Punjabi, Punjabi University, Patiala

[4] Omer Osman Ibrahim, & Yoshiki Mikami (2012). Stemming Tigrinya Words for Information Retrieval, Nagaoka University of Technology, Nagaoka, Japan.

[5] Atelach Alemu Argaw and Lars Asker, An Amharic Stemmer: Reducing Words to their Citation Forms, Stockholm University/KTH, Sweden

[6] Nega Alemayehu & Peter Willett (2003). The effectiveness of Stemming for Information Retrieval in Amharic. Journal: electronic library and information systems, Vol.37, num.4, pp.254-259

[7] Yitayal Abate (2014), Morphological Analysis of Ge'ez Verbs Using Memory Based Learning, Adis Ababa University

[8] Desta Berihu Weldegiorgis (2010), Design and Implementation of Automatic Morphological Analyzer for Ge'ez Verbs

[9] Brajendra Singh Rajput, & Dr. Nilay Khare (2015), A survey of Stemming Algorithms for Information Retrieval, IOSR Journal of Computer Engineering (IOSR-JCE: IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. VI (May – Jun. 2015), PP 76-80.

[10] Dinesh Kumar & Prince Rana (2010). Design and Development of a Stemmer for Punjabi, International Journal of Computer Applications (0975 – 8887), Volume 11– No.1.

[11] Harshali B. Patil, B. V. Pawar, & Ajay S. Patil (2016). a Comprehensive Analysis of Stemmers Available for Indic Languages: International Journal on Natural Language Computing (IJNLC) Vol. 5, No.1

[12] መምህር ደሴ ቀለብ (2002 ዓ.ም). ትንሳኤ ግዕዝ, በኢትዮጵያ ኦርቶዶክስ ተዋህዶ ቤተክርስቲያን በሰንበት ትምህርት ቤቶች ማደራጃ መምሪያ ማኅበረ ቅዱሳን፡ አዲስ አበባ

[13] Harshali B. Patil, B. V. Pawar, & Ajay S. Patil (Feb 2016). A comprehensive analysis of stemmers Available for Indic languages*: International Journal on Natural Language Computing (IJNLC) Vol. 5, No.1*

[14] R. Vijaya Lakshmi & IIDr. S. Britto Ramesh Kumar (2014). Literature Review: Stemming Algorithms for Indian and Non-Indian Languages. International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014). Vol. 2, Issue 3 (July - Sept. 2014)

[15] በኢትዮጵያ ኦርቶዶክስ ተዋህዶ ቤተክርስቲያን ሰንበት ትምህርት ቤቶች ማደራጃ መምሪያ ማኅበረ ቅዱሳን. by Ethiopian Orthodox Tewahedo Church Sunday Schools Department Mahibere Kidusan. Retrieved from http://eotcmk.org/a/category/%E1%8C%8D%E1%8B%95%E1%8B%9D-%E1%8B%AD%E1%88%9B%E1%88%A9/