Stemming Tigrinya Words for Information Retrieval

Omer Osman Ibrahim[1] *Yoshiki Mikami*[1]
(1) Nagaoka University of Technology, Nagaoka, Japan
sangor@gmail.com,mikami@kjs.nagaokaut.ac.jp

ABSTRACT

The increasing penetration of internet into less developed countries has resulted in the increase in the number of digital documents written in many minor languages. However, many of these languages have limited resources in terms of data, language resources and computational tools. Stemming is the reduction of inflected word forms into common basic form. It is an important analysis process in information retrieval and many natural language processing applications. In highly inflected languages such as Tigrinya, stemming is not always straightforward task. In this paper we present the development of stemmer for Tigrinya words to facilitate the information retrieval. We used a hybrid approach for stemming that combines rule based stemming which removes affixes in successively applied steps and dictionary based stemming which reduces stemming errors by verifying the resulting stem based on word distance measures. The stemmer was evaluated using two sets of Tigrinya words. The results show that it achieved an average accuracy of 89.3%.

KEYWORDS : Tigrinya, Stemming, Information Retrieval, Stop Word , Search Engine

Proceedings of COLING 2012: Demonstration Papers, pages 345–352,
COLING 2012, Mumbai, December 2012.

1 Introduction

Most Information Retrieval (IR) systems use inverted indexes for indexing and searching purposes. An inverted index contains a collection of terms and their corresponding occurrences in documents. However, a single word may occur in a number of morphological variants in the index, thus, increasing the size of the index and decreasing retrieval performance. Therefore, it is important to reduce different variants of words to their corresponding single form before indexing.

Stemming is a normalization step that reduces the morphological variants of words to a common form usually called a stem by the removal of affixes. Among the many normalization steps that are usually performed before indexing, stemming has significant effect in both the efficiency and the effectiveness of IR for many languages (Pirkola, A. & Järvelin, K. 2001). In addition to Information retrieval, stemming is also important in many other natural language processing application such as machine translation, morphological analysis and part of speech tagging. The complexity of stemming process varies with the morphological complexity of a natural language. Tigrinya belongs to the Semitic language family that includes languages like Arabic, Hebrew and Amharic. Those languages are highly inflected and derived. Thus, a hybrid approach would be a good choice to get correct stems so as to increase search precision. Dictionary verification applied after suffix stripping can further enhance the accuracy of the stemmer and it can also give meaningful stems. In this paper we report the development of stemming procedure for Tigrinya that combines rule based suffix stripping and dictionary based verification.

The rest of the paper is organized as follows. In section 2, we briefly review background information on the Tigrinya language and stemming approaches. In Section 3, we briefly present the Corpus used in this research. The stemming approach used is detailed in Section 4. The evaluation of the stemmer is discussed in Section 5. Finally, the conclusion of the research is given in Section 6.

2 Background

2.1 The Tigrinya Language

Tigrinya is a language spoken in the east African countries of Eritrea and Ethiopia. It is one of the two official languages of the State of Eritrea. It is also a working language of the Tigray region of Ethiopia. It is estimated to be spoken by over six million people both countries (http://www.ethnologue.com). Tigrinya is a Semitic language of the afro-Asiatic family originated from the ancient Geez language. It is closely related to Amharic and Tigre.Tigrinya is written in Geez script also called Ethiopic. Ethiopic Script is syllabic, each symbol represents 'consonant + vowel' characteristics. Each Tigrinya base character also known as "Fidel" has seven vowel combinations. Tigrinya is written from left to right. Tigrinya is a 'low resource' language having very limited resources: data, linguistic materials and tools.

2.2 Tigrinya Morphology

Tigrinya is a highly inflected language and has a complex morphology. It exhibits the root and pattern morphological system. The Tigrinya root is a sequence of consonants and it represents the basic form for word formation. Tigrinya makes use of prefixing, suffixing and internal changes to form inflectional and derivational word forms. Tigrinya Nouns are inflected for gender, number,

case and definiteness. For example,ሃገራት(hagerat) - countries, ተማሃራይ (temaharay) - male student, ተማሃራት (temaharit) - female student.Tigrinya adjectives are inflected for gender and number. For example, ጸሊም(Selim), ጸለምቲ (Selemti) meaning 'black' (masculine), 'blacks' respectively. Like other Semitic languages Tigrinya has rich verb morphology.Tigrinya verbs show different morphosyntactic features based on the arrangement of consonant (C) -vowel (V) patterns. For example, the root 'sbr' /to break/ of pattern (CCC) has forms such as 'sebere' (CVCVCV) in Active, 'te-sebre'(te-CVCCV) in Passive.

2.3 Related Work

Anjali Ganesh Jivani (Anjali Ganesh Jivani, 2011) classifies stemmers in to three broad categories: truncating, statistical and mixed. Truncating stemmers are related with removal of the affix of a word. They apply a set of rules to each word to strip the known set of affixes. The first stemmer following such algorithm was developed by Julie Beth Lovins in 1968 (Lovins,1968). A latter such stemmer was proposed by Martin Porter in 1980 (Porter, 1980). The major disadvantage of such stemmers is that they need large set of affixes and prior knowledge of the language morphology.

Yassir et. al. (Yassir, 2011) reported an enhanced stemmer for Arabic that combines light stemming and dictionary based stemming. In their research they included handling of multi-word expression and named entity recognition. They reported that the average accuracy of their enhanced stemmer was 96.29%. Sandip and Sajip (Sandip et. al. 2009) reported a rule based stemmer for Bengali that also uses stem dictionary for further validation . They used a part of speech(POS) tagger to tag the text and apply POS specific rules to output the stem. Not much work has been reported for Tigrinya language stemming compared to other languages. Tigrinya is a an under-resource language with very few computational work done on it. However, there is a pioneering work done on Tigrinya language morphological analysis and generation by Michael Gasser (Michael Gasser, 2009). Yonas Fissaha reported a rule based Tigrinya stemmer on his locally published thesis and reported an accuracy of 86% (Yonas Fissaha, 2011). We could not find any other publication on Tigrinya Stemmer following Hybrid approach.

3 The Corpus used

This work is part of an ongoing research to design a Tigrinya language Search Engine. As part of the research, we have crawled significant number of documents from the web using a language specific crawler that was specifically designed for Tigrinya web content. Our corpus includes a number of Tigrinya pages from different domains including news, religious, political, sport and so on. After cleaning the data, we generated a Tigrinya word lexicon of size 690,000 unique words. Finally, it was used to generate list of prefixes, suffixes and list of stop words for Tigrinya. The corpus is freely available for researchers in Tigrinya language and can be downloaded from the Eritrean NLP website (Eri-NLP: http://www.eri-nlp.com).

4 The Tigrinya Stemmer

The stemmer was developed to conflate different inflected forms of words into a common term so as to increase retrieval recall and decrease the size of the index. In addition to that, since different inflections of the same base word can have different stems, we as much as possible wanted to put such forms in to a common stem so as to increase the precision of retrieval.

The stemmer follows a step by step transformation of a word until it finally outputs the possible stem. Firstly, the word is checked for apostrophe suffixes attached. Next, any other punctuations or accents concatenated to the word are removed. The resulting string is then normalized for letter inconsistencies. It is then checked against stop word list. If not in the list, it is transliterated and passed to the prefix removal routine. The result is then forwarded to the suffix removal routine. The result is then further checked for single and double letter duplicates. Finally the resulting stem is verified in the stem dictionary. If an exact match is found it is taken as the final stem. If no exact match is found, possible candidates stems are selected from the dictionary based on the inclusion of the stems in the stripped string. If more than one stems are selected, the one closest to the stripped string is selected based on the string distance metric. If two or more of the candidate stems selected from the dictionary have the same distance from the stripped word, the stem with high frequency on the corpus is selected as a final result. Finally, the final selected stem is transliterated back to Ethiopic and displayed as an output.

4.1 Suffix Removal

In many Tigrinya web documents an apostrophe or other similar mark is placed at the end of words to add suffix or to show that a letter has been omitted. In most of these words the character represents the letter in the 'ኧ' series. For example, ይኹን'ምበር is meant to be read as ይኹን ኣምበር, ኣመሓላሊፉ'ሎ are meant to be read ኣመሓላሊፉ ኣሎ. In most cases, the character used is the apostrophe, the right single quotation mark or the left single quotation mark. Thus, a routine that handles such suffixes is added. It removes such suffixes in addition to any other punctuations concatenated to the word. Tigrinya uses Ethiopic punctuation which lies in the Unicode range of \u1360-\u1368. This makes sure that no non-letter characters remain in the string.

4.2 Normalization

The Ethiopic Script includes different letters that have the same sound in a language. The letters 'ሰ' (se) and 'ሠ' (se), letters 'ጸ' (Tse) and ፀ (Tse) are some examples. Although most Tigrinya writings have one of these forms, some writers use them interchangeably. In Eritrea the 'ጸ' series is used while in Ethiopia the ፀ series is used. Thus, a single Tigrinya word may exist in two different variations on many web documents. For example, መጽሂት(meShEt) and መፅሄት (me'ShEt) are two variants of the same word meaning 'pamphlet'. Such variant forms have negative effect on precision of retrieval. Thus, a routine is added to convert such variants in to a single form.

4.3 Transliteration

After the string is normalized, it is first checked against the stop word list. Stemming a stop word is a useless process because stop words are not indexed. Hence, only those words that are not on the list are further processed by the stemmer. The transliteration step converts the Ethiopic string to Latin. We need this because the Ethiopic Script is Syllabic where both consonants and vowels are joined together in a Symbol. Thus, without transliteration it is very difficult to make word analysis such as affix removal or string comparison. By transliterating we convert the Tigrinya Symbols to their corresponding consonant + vowel combinations which makes it suitable for affix removal. For example, the word ሃገራት /hagerat/ meaning 'Countries' is composed of the stem ሃገር /hager/ and the number marker suffix ኣት/at/. In this case, the letter ኣ'a' which is part of the suffix is fused in the letter ራ/ra/.If the suffix removal is done before Romanization, either the ት/t/ or ራት/rat/ can be removed which leads to wrong stem. Romanizing the word to /hagerat/ makes it convenient to delete the suffix /at/ and retrieve the stem /hager/.We used the

transliteration conventions of SERA - 'System for Ethiopic representation in ASCII' with few exceptions (Firdyiwek and Yacob, 1997).

4.4 The Affix Removal

This is the step where most of the inflections are removed. Tigrinya affixes include prefixes which are placed before a word and suffixes placed after the stem of a word. A Tigrinya word contains zero or more prefixes and zero or more suffixes. Our affix removal algorithm depends on a list of prefixes and suffixes. For this purpose, we have manually compiled a 153 set of prefixes and 204 set of Suffixes from our corpora mentioned in Section 3. The main disadvantage of affix removal strategies is that they require the knowledge of the language's morphology beforehand. Given the complex morphology of Tigrinya words, it is not easy to come up with a general set of context sensitive rules that govern the affix removal process. However, we have constructed a basic set of affix removal rules by studying various Tigrinya words with different parts of speech. The affix removal algorithm is similar to the one used by (Alemayehu and Willet ,2002) on Amharic. It iteratively removes the longest matched affix from a word by considering the minimum stem length. It also considers the length of the word to be stemmed so that to decrease over-stemming. The shortest Tigrinya word consists of two consonants. Hence, for a word to be further processed, it should at least have three consonants.

4.5 Duplicate Consonant Handling

Some Tigrinya verbs are derived by doubling of a consonant form. This words are of two types: those words containing consecutive consonants of the same form and those words containing one consonant form followed by the same consonant of different form. For example, consider the word ተሰሓሓቢ /teseHaHabi/. The Romanized form of the word shows that the consonant 'H' occurs consecutively. In the consonant-vowel fused form, forth form ሓ (Ha) is doubled in the word. In this specific case, after the prefix step removes the prefix 'te', the remaining word ሰሓሓቢ /seHaHabi/ is handled by removing the double consonant and changing it to the sixth form to give ሰሕቢ. For the second types consider a word of the form XYC_6C_4WZ where each of the letters represent a consonant-vowel Ethiopic form. The sixth order(C_6) is followed by the fourth order(C_4). In such forms, C_4 is deleted. For example, the word ምንግጋር /mnggar/. The affix removal step gives us the word ንግጋር /nggar/ and it is reduced to ንግር /ngr/. By studying the patters in many such words, we have constructed a set of rules for changing such forms.

4.6 Dictionary Verification

This step is introduced to further increase the accuracy of the stemmer. It helps put different stems of the same word into a single form. The Tigrinya stem dictionary employed was constructed manually from the corpus and it contains a stem, its Romanized form and its frequency in the corpus. Firstly, the resulting string is searched in the stem lexicon for an exact match. If found, it is returned as a final stem and the process is terminated. This is done as a means of validation of the correctness of the stem. If an exact match is not found, verification is done as follows: all the stems that are contained in the stripped string are selected as candidate stems. To select the most likely stem, we introduce string distance metric between the string and the potential stem candidates. The candidate with minimum distance from the word is given highest priority. We use the Levenshtein Edit distance (Levenshtein, 1966). The Levenshtein distance (commonly known as Edit Distance) between two strings is the minimum number of

operations needed to change one string into another. The edit distance between each candidate stem and the string is calculated. Those candidates that are above a certain edit distance are removed from the candidates list. This is important because sometimes stems which are unrelated to the word may be selected as candidates stems. In order to avoid the wrong matching of such stems, the candidates which are very far from the word are removed from the selection. This ensures that only the stems which are related to the word get considered and thus avoids inconsistent matching which would lead to a wrong stem. The remaining candidates are ranked on the basis of their edit distances. Finally, the stem with minimum distance from the string is selected as a final result of the stemmer. If two or more of the candidates have equal edit distance, their frequency on the corpus is used and the more frequent stem is selected as final result.

5 Evaluation and Discussion

To evaluate the design of the stemmer, it was implemented in C# programming language. The stemmer was then tested using two sets of words. The purpose of the evaluation was to calculate the accuracy of the stemmer. The set of words used for evaluation were selected from different domains. The first sample was extracted from an Eritrean news paper (Hadas Eritra) and the first 1200 unique words were selected. The second sample was extracted from an online Tigrinya Bible. Similarly the first 1300 unique words were selected. The test samples were supplied to the stemmer and the results were manually checked. On the first sample the stemmer achieved an accuracy of 89.92% while the accuracy on the second sample was 88.6%.

The stemmer produces some over-stemmed and under-stemmed words. However, the accuracy rate was acceptable for the purpose of our work which is to develop a Lucerne Analyzer for Tigrinya. The use of the dictionary checking method was the core reason for enhancement. The stemming errors during the affix removal step were fixed by the dictionary based stemmer. However, the dictionary based stemming is not be applied to words that are not in the stem lexicon. The repetitive consonant handling and the apostrophe suffixes steps contribute much less than the other steps because the number of such Tigrinya words is small. We can easily get the root of a Tigrinya word from its stem by deleting all vowels from the stem to get a sequence of consonants.

6 Conclusion and Future Work

In this paper we presented a stemmer for the highly inflected Tigrinya language. The stemmer was designed for the purpose of representing different forms of a Tigrinya word with single form to enhance the effectiveness of Tigrinya Information Retrieval applications. Although the overall accuracy was satisfactory, it can further be enhanced for higher accuracy and meaningful stems.

Tigrinya inflections vary with part of speech (POS) of the words. Currently there is no POS tagger for Tigrinya. Introducing a POS tagger would help apply different affix removal rules on the basis of POS of a word. Adoption of more context sensitive rules would also give better results. The distance metric used gives the same priority to both consonants and vowels. However, the root of Tigrinya words consists of only consonants. Thus, adopting a distance metric that considers this would also increase the matching rate of the dictionary based stemmer. Further study on the above points will be done in the future to improve the accuracy of the stemmer. The stemmer will be used to study the effectiveness of stemming in Tigrinya information retrieval and to develop a Tigrinya language Search Engine.

References

Pirkola, A. & Järvelin, K. (2001) :*Morphological typology of languages for IR*. Journal of Documentation, 57 (3): 330-348.

Ethnologue, Languages of the World (2011) : http://www.ethnologue.com.

Michael Gasser. (2009). *Semitic morphological analysis and generation using finite state transducers with feature structures*. In Proceedings of the 12th Conference of the European Chapter of the ACL, pages 309–317, Athens, Greece.

Anjali Ganesh Jivani et al, (2011). *A Comparative Study of* Stemming *Algorithms* , Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938.

J. B. Lovins,(1968). *Development of a stemming algorithm*, Mechanical Translation and Computer Linguistic., vol.11, no.1/2, pp. 22-31.

Porter M.F (1980). "*An algorithm for suffix stripping*". Program:14, 130-137.

Krovetz Robert(1993). "*Viewing morphology as an inference process*". Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval: 191-202.

Yasir Alhanini, Mohd Juzaiddin Ab Aziz, (2011).*The Enhancement of Arabic Stemming by Using Light Stemming and Dictionary-Based Stemming*, Journal of Software Engineering and Applications, 4: 522-526, ISI Wos, DBLP.

Sandipan Sarkar and Sivaji Bandyopadhyay,(2009). *Study on Rule-Based Stemming Patterns and Issues in a Bengali Short Story-Based Corpus*. In ICON.

Yonnas Fissiha (2011):*Developement of Stemming algorithm for Tigrinya,* URL: http://etd.aau.edu.et

Alemayehu, N., Willett, P. (2002). *Stemming of Amharic words for information retrieval. Literary and Linguistic Computing* 17(1), 1-17

Firdyiwek, Yitna and Daniel Yaqob. (1997). *The system for Ethiopic representation in ASCII* URL: citeseer.ist.psu.edu/56365.html.

V. Levenshtein (1966). *Binary codes capable of correcting deletions, insertions and reversals*. Soviet Physics Doklady, 10(8):707–710.

Eri-NLP (2012): *Eritrean Language Research Website*, URL: http://www.eri-nlp.com/resources