International Journal on Data Science and Technology
2018; 4(3): 79-83
http://www.sciencepublishinggroup.com/j/ijdst
doi: 10.11648/j.ijdst.20180403.11
ISSN: 2472-2200 (Print); ISSN: 2472-2235 (Online)

Development of Longest-Match Based Stemmer for Texts of Wolaita Language

Girma Yohannis Bade[1], Hussien Seid[2]

[1] Department of Computer Science,Wolaita Sodo University, Wolaita, Ethiopia

[2]Department of Computer Science and IT, Arba-Minch University, Arba-Minch, Ethiopia

Email address:
girme2005@gmail.com (G. Y. Bade), lehussien@gmail.com (H. Seid)

*Corresponding author

To cite this article:
Girma Yohannis Bade, Hussien Seid. Development of Longest-Match Based Stemmer for Texts of Wolaita Language. *International Journal on Data Science and Technology*. Vol. 4, No. 3, 2018, pp. 79-83. doi: 10.11648/j.ijdst.20180403.11

Received: May 19, 2018; **Accepted**: July 5, 2018; **Published**: July 30, 2018

Abstract: This research presents design, experiment and development of longest-match based Stemmer for Wolaita texts. The objective of this paper is to conflate the variants of Wolaita text words into its stem with better accuracy, using Longest-Match based approach. To help the researcher how to compile the possible combination of suffixes, the deep analysis of Wolaita word morphology has been made. For data preprocess and implementation, C# programming language is used. After preprocessing, 12789 unique words are reserved to experiment this research. Out of these unique words, 1200 words are randomly selected earlier and kept separate for testing purpose. Then the developed stemmer was tested using Paice's actual error counting method. The output on that test dataset has showed 91.84% accuracy over actual manually stemmed words. The obtained result shows that the rule based longest match approach is promising for stemming Wolaita language texts.

Keywords: Stemmer, Natural Language Processing, Morphology, Longest-Match

1. Introduction

Omotic languages are a group of close to 30 languages which are spoken in the south west of Ethiopia around the Omo river. Among these, the 28 Omotic languages are classified into northern and southern sub-families [1]. Wolaita language is one of the Northern Omotic languages that is spoken in the Wolaita Zone and some other parts of the Southern Nations, Nationalities, and People's Region of Ethiopia. The language has around 3.3 million of native and dialective speakers. The Latin script is being used since 1993 to write Wolaita texts [2]. As a result, the publications of textbooks and other reference materials like literatures, newspapers, and magazines have been increasing over the year; and a significant number of people are able to read and write Wolaita scripts. The language is also serving as a medium of instruction in primary school and is offered as a subject in secondary school, and a program in Wolaita Sodo University.

Wolaita language is one of a morphologically rich language in Ethiopia. Affixation and compounding are the two common ways of forming words in Wolaita [4]. The words formed in either way are derivate or inflect.

2. Statement of the Problem

There is no Wolaita computational linguistics, and is quite difficult to follow the same stemming pattern and rules of others. Getting out the stem for the particular words may need the language experts (elders) in that language which again may reveal the inconsistences and time consuming to stem more words.

In addition, Information Retrieval (IR) system, Automatic text processing system, Text summarization, spell checkers and others highly uses Stemmer.

3. Objectives of the Study

The objective of this study is the development of longest-match based stemmer for Wolaita texts. To achieve this objective, the following specific activities were done:

1. Wolaita text corpus was collected;
2. A possible suffixes in Wolaita text was compiled;

3. A prototype stemmer is developed for Wolaita text;
4. The performance of stemmer was tested to see how it compress words.

4. Significances of the Study

1. Can help as computational linguistic: - as in this time the language linguists gets disappear, this the developed stemmer will help the language owners as computational linguists.
2. For Information Retrieval system: - The applications of stemmer in information retrieval is increasing recall without decreasing precision, because both document indexes and queries use stems.
3. Its applicability to other natural language processing applications such as word frequency count, natural language generator, spell checkers, and word parsing devices. Second, the system can be served as computational linguistic expert. Hence, an accurate Wolaita stemmer can be used for various Wolaita Language technologies.
4. The collected corpus as a resource: - the collected electronic data is very useful and can be easily accessible for which the application requiring it.

5. Research Methodology

Information Gathering

The problem identification was made by making discussion with language experts and analyzing written document. To have more insight on the research, a thorough discussion was made with linguistic experts of the Wolaita language. The discussion helped to gain the most important concept of how the stem of the word in Wolaita text can be generated. This was used as primary input in addition to secondary information that has been analyzed the written document in the course of the study [6].

Stemming Approach

Affix removal which is one of automatic conflation method has two approaches to remove affixes; longest-match and iterative [3]. In this study, longest-match approach was adopted. Longest-match approach, on contrary to iterative approach, involves only a single pass, i.e., if more than one suffix matches the end of the word, the longest one is removed. Even though the Longest-match approach requires the compilation of all possible combinations of suffixes; it has less computational complexity because the arrangement of suffixes in suffix list are in their decreasing order of length and has less time complexity because it involves in single pass of the suffix match. In addition, longest-match approach is often easier to program [5, 10].

Test dataset collection

There is no public balanced text corpus for Wolaita texts. For testing purpose, totally 46,180 words of datasets were collected from three different domains (Wolaita-English Dictionary, Bible in Wolaita, and Wolaita Text Books from grade 7-8). After preprocessed, 12789 unique white space separated words were left and which included the inflected and derivate form of words according to their tense, number, gender and moods. Among these 1200 words (around 10%) of texts were randomly selected and put in another file list to test the performance of stemmer.

Wolaita Suffix List Compilation

Unlike Semitic languages in Ethiopia [8], Wolaita is a language that only depends on suffix to form different forms of a given word [1, 9]. Thus the radicals of some words are very long. Among the suffixes in suffix list, the maximum and minimum length of suffix is 16 and 1 respectively. To reduce computational complexity, the suffixes in the suffix list are also arranged in descending order. Totally 803 possible suffixes were compiled for this study. However, the following table only shows 81 basic Wolaita suffixes. They all should be in small-case because they came at the end of another word unless they act as stop word themeselve. For example “as-a” to mean person, stem is “as-“and suffice is ‘-a’.

Table 1. *List of 81 Wolaita Basic suffixes.*

-ata	-okk	-ibe	-asa	-awu	-ayo	-eti	-an
-gaa	-oow	-ana	-uni	-uge	-eta	-an	-ua
-ida	-era	-iba	-ude	-ide	-ade	-is	-os
-ara	-eto	-adi	-ike	-atu	-oos	-ok	-oy
-aas	-ees	-idi	-ta	-yoo	-ayi	-ow	-ii
-awu	-iss	-ena	-ays	-ota	-ini	-ey	-ee
-ussi	-ani	-ati	-iis	-era	-ura	-ay	-i
-iya	-ada	-eni	-isi	-usa	-ido	-aa	-a
-uwa	-eda	-oti	-aan	-iin	-out	-on	-e
-iyo	-ida	-ona	-eyo	-ibo	-etu	-en	-u
-o							

System prototyping tools

To develop the system, C# programming language was used. Because, C# being a. NET language supports language interoperability. In addition, through C# we can easily call Windows API function and access COM components. Hence, the researcher used Visual Studio 2010 Express as a tool and C# programming language as one of its components.

6. System Design and Performance Analysis

System design

The prototype stemmer which developed in this study starts stemming process as follows. First, it loads the words to be stemmed from the file, checks the size of radicals, if the size of radical greater than or equal to two, then it opens the file of suffix lists. However, suffixes in the suffix list are arranged in their decreasing order; so it starts scanning at top suffix in suffix list. If the match found, the stemmer strips those endings from the words obeying available conditions and returns stem. The following algorithm shows the details in step ways.

1. LOAD the word to be stemmed from file
2. Check the size of radical

If the radicals are greater or equal to two

Go to 3
ELSE
Go to 5
3. OPEN suffix file lists and READ it
IF match exists
Go to 4
ELSE
Go to 5
4. Check the Exception
IF Exception met
Go to 5
ELSE
Remove the suffix from word
Got to 6
ELSE
Go to 5
5. Return the word
6. Return the stem
7. IF end of file not reached
Go to 1
ELSE
Stop processing
8. END

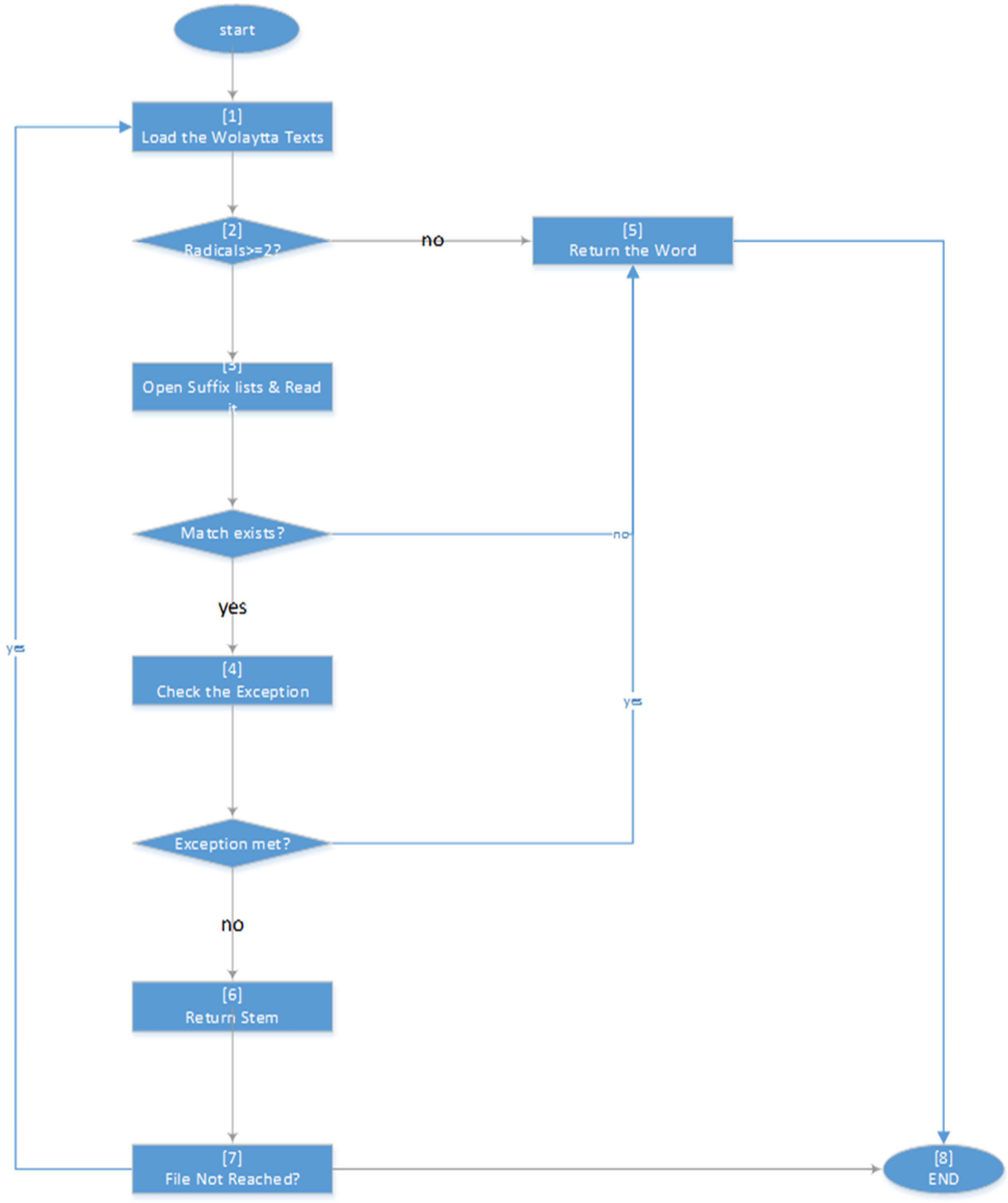

Figure 1. *Hypothetic Wolaita longest-match stemmer.*

Performance Analysis on sample datasets

Table 2. *Shows the performance of stemmer comparing stems stemmed by the stememer developed over manual with corresponding remarks.*

Un-stemmed words	System stemmed	Manually stemmed	Remark
Geeliichchida	geeliichch-	geel-	Under stemmed
de'ikkokka	de'-	de'-	Correct
Ammanoppite	amman-	amman-	Correct
Amaridaagaa	amar-	amar-	Correct
Qoodiiddi	qood-	qood-	Correct
Uusuntta	uusunt-	uusuntt-	Over stemmed
Pholqqu	pholqq-	pholqq-	Correct
Xishe	xish-	xish-	Correct
Qonttatettay	qontt-	qonttatett-	Over stemmed
Gallassawu	gallass-	gallass-	Correct
Oyqqiis	oyqq-	oyqq-	Correct
Manttaa	mantt-	mantt-	Correct
Keeri	keer-	keer-	Correct
Ammanekketa	amman-	amman-	Correct
Nunatetta	nun-	nun-	Correct
Pito	pit-	pit-	Correct
na'eera	na'-	na'-	Correct
Sissana	s-	s-	Correct

7. Result and Discussion

At the beginning, 1200 words of the test datasets were randomly selected from total unique sample datasets in order to predict the performance of the stemmer in the real world data. The words included in test datasets are verbs, nouns, adjectives, adverbs, compounds, and irregulars and each of which have inflectional and derivational form. Moreover, these test datasets were manually stemmed. And on the other hand, the stemmer was run on same these datasets. Then the software that was developed for test purpose compared this manually stemmed datasets with that of the system stemmed ones and generated 91.84% accuracy. I.e. 8.16% of words are incorrectly stemmed. Of which 2.08% are under stemmed and 6.08% are over stemmed. The following table below shows the detail.

Table 3. *Summarizes the result of the stemmer.*

Number of incorrectly stemmed=98 i.e.,(98/1200)*100=8.16%	**Under stemmed=25 i.e., (25/1200)*100=2.08%**
	Over stemmed=73 i.e.,(73/1200)*100=6.08%
Nº of Correctly stemmed=1102	(1102/1200)*100=91.84%
The accuracy of the stemmer	(Nº of Correctly stemmed/total)*100=91.84%

The main advantage of stemming is reducing one class of a variety of words to a single stems, the process is called word compression. In terms of compression, i.e., reduction of dictionary size, using Paice [7] formula:

$$C = 100 \times \frac{(T_W - D_S)}{T_W} \quad (1)$$

Where, C is the compression value (in percentage), T_w is the number of the total words and D_s is a distinct stem after conflation. Accordingly,

1. Size of the test datasets (Tw)= 1200
2. Number of distinct stems (Ds) = 755

This 755 distinct stems are obtained by removing duplicates from the output of the stemmer because; before running into the system we have no duplicate terms. Hence, the percentage of compression for Wolaita text based on test datasets, applying the above formula becomes:

C= 100 * (1200 - 755)/ 1200 = 100*(445/1200) = 37.08%.

This indicates that after stripping the suffixes, 445 words of duplicates have been found and then removed. Then 755 distinct words are remained, and this also shows the balanced distribution of test datasets. Thus, the compression rate of test datasets from 1200 unique words to 755 distinct words is 37.08%.

Generally, if the words returned from the stemmer as it is, it must be personal pronouns (stop words) such as {a, i, o} otherwise every word should be stemmed correctly or incorrectly. Because every Wolaita words end with one of the five vowel phonemes and they are present in the suffix list as suffixes, at least one character of suffix will be matched and removed.

8. Conclusion

Stemming is important for highly inflected languages like Wolaita for many applications that require the stem of a word. The longest match approach of affix removal procedure has been employed in this study. According to the evaluation result of the stemmer on test datasets, about 91.84% of accuracy was shown. When comparing with the accuracy of the previous work (iterative approach) the stemmer developed in this study was showed 4.94% of improvements. The stemmer developed in this study can

compress the vocabulary of Wolaita in 37.08%. When the compression rate increases the strength of the stemmer increases. Thus, the longest match is the most appropriate method to conflate Wolaita texts than iterative.

Table 4. *The comparison between the previous work and current work.*

	Approach used	Size of test datasets	stemmer accuracy	Vocabulary reduction in test datasets
Previous Work	Iterative	884	86.9%	41%
Current Work	Longest match	1200	91.84%	37.08%

For further improvement of the stemmer, a deep analyses on compounding and irregulars words should be made.

The stemmer has to be tested with large amount of texts to prove its real performance. To succeed this we need to apply Wolaita stemmer in a web search engine, which retrieves information from Wolaita texts.

References

[1] Wardhaugh, R. Introduction to Linguistics. New York: McGraw-Hill Book Company, (1977).

[2] Lemma Lessa. “Development of stemming algorithm for Wolaita text.” M. Sc. Thesis, Addis Ababa University, Department of Information Science, Addis Ababa,(2003).

[3] Salton, G. & McGill, N. “Introduction to Modern Information Retrieval”. New York: McGraw-Hill, (1983).

[4] Liddy, E. “Enhanced text retrieval using natural language processing.” Bulletin of the American Society for Information Science, 24, PP. 14-16, (1983).

[5] Schinke, R, et al. "A Stemming Algorithm for Latin Text Databases." In Journal of Documentation. 52(2), PP. 172 – 187, (1996).

[6] Lamberti, Marcello and Sottile, Roberto. ”The Wolaita Language. Koln: Rudiger Koppe Verlag.”, (1997).

[7] Paice C. D. “An Evaluation Method for Stemming Algorithms”. ACM SIGIR Conference on Research and Development in Information Retrieval. 1994, 42-50.

[8] McGregor, W., (2009). Linguistics: An Introduction. London: Continuum International Publishing Group.

[9] Debela T, Ermias. Designing a Rule Based Stemmer for Afaan Oromo Text. International Journal of Computational Linguistics (IJCL), Volume (1): Issue (2), October 2010.

[10] Dawson J. L., 1974: "Suffix removal and word connation," *ALLC Bulletin*, **2**(3), 33-46.