Implemented Stemming Algorithms for Information Retrieval Applications

Wubetu Barud Demilie
Department of Information Technology, Wachemo University, Hossana, Ethiopia, P.O. Box 667

Abstract
Now a day's text documents are advancing over internet, e-mails and web pages. As the use of internet is exponentially growing, the need of massive data storage is increasing from time to time. Normally many of the documents contain morphological variables, so stemming which is a preprocessing technique gives a mapping of different morphological variants of words into their base word called the stem. Stemming process is used in information retrieval applications accordingly as a way to improve retrieval performance based on the assumption that terms with the same stem usually have similar meaning. To do stemming operation on bulky documents, we require normally more computation time and power, to cope up with the need to search for a particular word in the data. In this paper, various stemming algorithms are analyzed with the benefits and limitation of the recent stemming methods or approaches.

Keywords: - Natural Language Processing Applications, Information Retrieval, Information Retrieval Applications (IRAs), Stemming Approaches
DOI: 10.7176/IKM/10-3-01
Publication date: April 30th 2020

1. Introduction

In all Information Retrieval applications, the main thing is to improve recalls and precisions accordingly. A recall increasing method which can be advantageous for even the simplest Boolean retrieval systems is stemming. Stemming is a preprocessing footstep in text mining applications as well as a very common requirement of natural language processing functions. In fact, it is very important in most of the information retrieval applications. The main purpose of text stemming is to reduce different grammatical forms / word forms of a word like its noun, adjective, verb, adverb etc. to its root/stem form(Haroon 2018) (Bade and Seid 2018)(Adege and Manie 2017).

Information discoverer who is looking for texts say "cats" is probably interested in the texts which consist of the term "cat" only. The capacity of the search database has increased in the last few years, so in order to meet the challenge of real time search natural language application algorithms speed up required. Those texts typically consist of many different syntactic variants for example connected, connect, connecting, connection, connectedly, connectedness, connectively, connectional, connective, connectable (adjective), connector (noun) all are derived word of root word "connect"(Tesfaye 2010)(Tesfaye n.d.).

The conventional methodology used to mine data for some user query is to search the documents present in the given corpus/dataset word by word for the given query. This method is very time taking and it may leave some of the comparable documents of equivalent nature. Thus, to avoid these situations, stemming has been broadly used in various information retrieval applications to maximize the extracting accuracy indexing terms. In all stemming algorithms, the main purpose is to choose maximum representative feature, scopes base on correspondence/similarity measurement. For example: Consider the following diagram to clearly understand its main goal in IRAs (Argaw and Asker 2007).

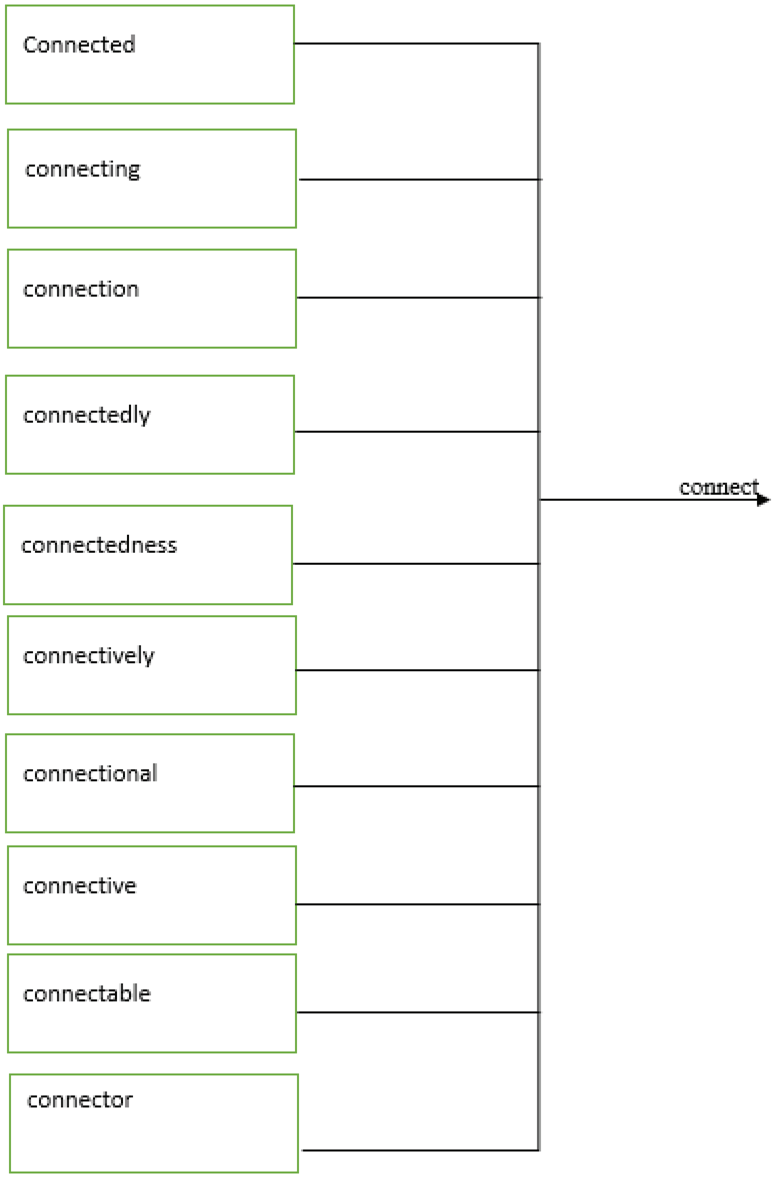

Figure 1:derived word of root word connect

From the above diagram, we can understand that the derived words connected, connecting, connection, connectedly, connectedness, connectively, connectional, connective, connectable and connector are converted to root word "connect", through which not only retrieval performance improve but also storage can be adjusted in some definite applications.

2. Working Principles of Stemmer

It has been seen that most of the times the morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of information retrieval applications. Since the meaning is equivalent but the word form is dissimilar(Ferro 2002)(Sever and Bitirim n.d.). It is necessary to categorize each word form with its base form. To do this, a variety of stemming approaches have been developed by different scholars. Each approach attempts to change the morphological variants of a word like connection, connected, connecting etc. to get mapped to the word "connect". Some approaches may map them to just differently, but that is permitted as long as all of them map to the same word form or more commonly known as the stem form.
Thus, the important terms of a request or file are symbolized by stems rather than by the original words. The idea is to decrease the total number of dissimilar terms in a file or a query which in turn will decrease the processing time of the final output for any kind of information retrieval applications (Demilie 2019).

3. Conflation Approaches

According to (Ismailov et al. 2016) there are to stemming approaches. The first stemming approach is stemming which simply means of context free with the main objective of identifying affixes and removing them. The second stemming approach is lemmatization. In lemmatization, the developer has to have a good knowledge of the language and its grammatical rule.

It also requires a dictionary look up; therefore, it is more complex than stemming. However, in lemmatization more accurate and precise result is expected.

For example, a word 'better' has a lemma 'good'. These types of words cannot be solved in basic stemming approaches unless it uses dictionary look-up table. To achieve stemming, there are different types of stemming approaches that are available for different languages, which differ in terms of performance and accuracy. The most basic stemmer looks up the inflected form in a lookup table. The benefits of this kind of stemming approach is simple, fast, and easy to handle different exceptions. However, this approach also has many weaknesses; for example, the inflected word forms have to be explicitly and clearly listed in the table. Otherwise, new words or unfamiliar words (the words are not included in table list) will not be stemmed, even if words are perfectly regular (e.g. iPads ~ iPad), and hence the table could be big (Ismailov et al. 2016).

According to (Haroon 2018) the massive data is increasing exponentially on web and information retrieval systems and the data retrieval has now become challenging. Stemming is used to produce meaningful terms by stemming characters which finally result in accurate and most relevant results. The core purpose of stemming approach is to get useful terms and to reduce grammatical forms in morphological structure of languages(Anjali and Jivani 2011; Otair 2016).

In order to perform stemming activities, we have to conflate a word to its different deviations.

Conflation approaches which are used in stemming application are shown in figure 2.

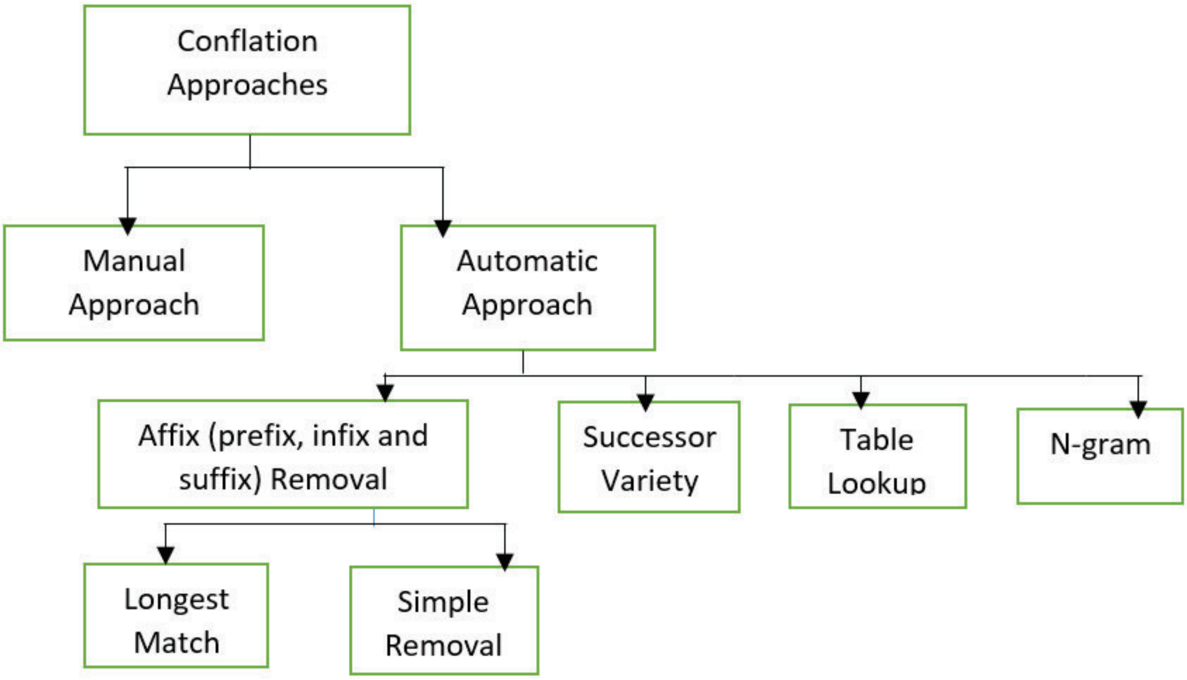

Figure 2:Conflation approaches

3.1. Affix (Prefix, Infix and Suffix) Removal

The affix removal approaches eradicate prefixes, infixes and suffixes from word in order to decrease word into common base forms. Most of stemmer used this type of approach for conflation. These approaches depend on two ideologies one is iteration, which removes strings in each order class once at a time, starting at the end of a word and going towards its beginning. Not more than one match is allowed in a single order class. The suffix is added to a word in any arbitrary order, that is, there exist order classes of suffix. The longest match is second type in which within any given class of endings, if more than one ending gives a match then longest match should be eliminated (Adege and Manie 2017)(Anon 2020).

3.2. Successor Variety Approach

According (Sousa and Castro n.d.) to successor variety is one of the stemming approaches in natural language processing applications including especially, in information retrieval processing systems. In this approach, the successor variety of a string is the number of different characters that follow the string in words in a corpus (the body of text). The successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. Successor variety stemmer does not require preparation of suffix lists and removal rules, and hence can be adapted to changing text collection (Anon 2009)(Stein and Potthast n.d.).

According to (Ehret et al. 2014) successor variety approach uses the frequencies of letter sequences in a body of text as the basis of stemming. In less formal terms, the successor variety of a string is the number of different characters that follow it in words in some body of text. Consider a body of text consisting of the following words, for example, back, beach, body, backward and boy. To determine the successor varieties for 'battle', for example, the following process would be used. The first letter of battle is 'b'. 'b' is followed in the text body by four characters: 'a', 'e', and 'o'. Thus, the successor variety of 'b' is three. The next successor variety for battle would be one, since only 'c' follows 'ba' in the text. When this process is carried out using a large body of text, the successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. At this point, the successor variety will sharply increase. This information is used to identify stems.

3.3. Table Lookup Method

Table lookup method is done by looking at the table where the term stems and their matching stored. Term from queries and indexes could be stemmed by then a lookup table. If we use B-tree or hash table lookup then such would be fast, but there is a problem of storage overhead for such table(Bellovin and Rescorla 2005).

3.4. N-Gram Method

For calculating this association measures we use Dice's coefficient (Anon n.d.).
For example, the terms information and informative can be broken into digrams as follows.

- information => in nf fo or rm ma at ti io on

unique digrams = in nf fo or rm ma at ti io on

- informative => in nf fo or rm ma at ti iv ve

unique digrams = in nf fo or rm ma at ti iv ve

Thus, "information" has ten digrams, of which all are unique, and "informative" also has ten digrams, of which all are unique. The two words share eight unique digrams: in, nf, fo, or, rm, ma, at, and ti.
Once the unique digrams for the word pair have been recognized and counted, a similarity measure based on them is calculated. The similarity measure used is Dice's coefficient, which is defined as:

$$S = \frac{2C}{A + B}$$

where "A" is the number of unique digrams in the first word, "B" the number of unique digrams in the second, and "C" the number of unique digrams shared by "A" and "B". For the above example,

$$S = \frac{2*8}{10+10} = 0.80.$$

Such similarity measures are determined for all pairs of terms in the database. Once such similarity is calculated for all the word pairs, they are clustered as groups. The value of Dice coefficient gives us the hint that the stem for these pair of words lies in the first unique 8 digrams.

4. Types of Stemming Approaches

Mostly stemming approaches can be classified into two types, rule based and statistical. Each type has its own way to discover a stem. Rule based stemmer encodes language specific rules, whereas, statistical information from a large corpus/ dataset of a given language to learn the morphology clearly.

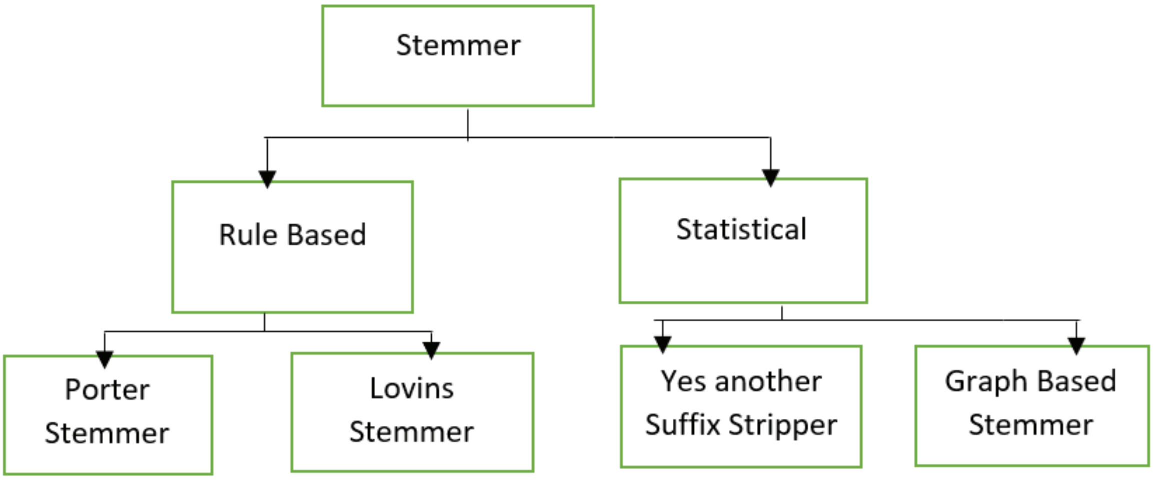

Figure 3:types of stemming approaches

4.1. Rule Based Stemmer

The rule based approach is implemented by different researchers and composed of two parts: a rule based light stemmer, and a patter based infix remover. The rule based light stemmer removes prefixes and suffixes form the word according to specific rules. The pattern based infix remover removes infixes from the word according to specific patters. This approach is named here rule based approach (Tesfaye 2011)(Synthesizer and Abeshu 2013)(Samuel et al. 2018)(Adege and Manie 2017)(Al-nashashibi, Neagu, and Yaghi 2010)(Anon 2020).

4.1.1. Porter Stemmer

In standard porter stemmer, there are five steps and sixty conditions. There is many modification "s" of standard procedures and its used for English file processing. General rule of eliminating suffix is given as:

(Condition)S1 ➡ S2

Whenever condition is fulfilled suffix "S1" is replaced by suffix "S2". The order of consonants(C), vowel (V) and consonants (C) is counted as measure function (m) in porter stemmer. When the measuring purpose is greater than one, then only convinced condition will be applied (Kraaij n.d.).

4.1.2. Lovins Stemmer

In Lovins stemmer, there are 29 conditions, 35 transformations rules and it perform lookup on a table of 294 endings. Here stemming encompasses of two phases. In the first phase, the stemming procedure repossesses the stem of a word by eliminating its longest possible ending by matching these ending with the list of suffixes stored in computer and in the second phase spelling exception are handled. For example, the word "absorption" is derived from the stem" abort" and "absorbing" is derived from the stem" absorb". The problem with the spelling exception arises in the above case when we try to match the two words "absorpt" and "absorb". Such exceptions are handled very carefully by introducing recording and partial matching techniques in the stemmer as post stemming procedures.

Rule dependent stemmer is fast in nature means calculation time used to find a stem is less. The retrieval result for English by using a rule dependent stemmer is reasonable, but the problem associated with rule based is one need to have extensive language expertise to make them.

4.2. Statistical Stemmer

The statistical stemmer is good alternative to rule based stemmer and does not involve language expertise.

They use statistical information from a large corpus/dataset of a given language to learn the languages morphological structure. Statistical language processing has been successfully used to increase the performance of information retrieval systems in the absence of extensive linguistic resources for some language.

4.2.1. Yet Another Suffix Stripper (YASS)

Yet another suffix stripper is one of statistical based language independent stemmer and its performance can be equated with mutually rule based stemmers in term of average precision. In this method, a set of string distance measure is used. The string distance measure is used to check the similarity between the two words by computing the string distance between two strings. The distance function maps a pair of string "an" and "b" to a real number "r", where "a smaller value of "r" indicates greater similarity between "an" and "b". The main reason for estimating this distance is to find the longest matching prefix from the given alternatives.

4.2.2. Graph Based Stemmer (GRAS)

It is a graph based language independent stemmer for information retrieval application processes. Extracting effectiveness, simplification and low computation costs are the features of graph based stemmer.

5. Errors of Stemming

There are fundamentally two kinds of fault in stemming approaches, namely over stemming and under stemming. Over stemming occurs when two words which have dissimilar root word are changed to the identical base term, which is also identified as a false positive. In under stemming two words which have similar root are not stemmed to the same base term, which is also called as false negative.

6. Conclusion

I have studied a variety of implemented stemming approaches for information retrieval applications and got to know that stemming appreciably increases the retrieval effectiveness and results for both rule dependent and statistical approaches. It is also useful in decreasing the size of index files and feature set or feature as the number of words to be indexed are condensed to mutual forms called stems. The performance of statistical stemmers is far superior to some well-known rule based stemmers but time consuming. Rule reliant on stemmers like porter stemmer is good choice for English document processing but its language dependent.

References

Adege, Abebe Belay, and Yibeltal Chanie Manie. 2017. “DESIGNING A STEMMER FOR GE’EZ TEXT USING RULE BASED APPROACH.” 8(1):1574–78.

Al-nashashibi, May Y., D. Neagu, and Ali A. Yaghi. 2010. “Stemming Techniques for Arabic Words : A Comparative Study.” *Stemming Techniques for Arabic Words : A Comparative Study* (I):270–76.

Anjali, Ms, and Ganesh Jivani. 2011. “A Comparative Study of Stemming Algorithms.” 2(6):1930–38.

Anon. 2009. “AN EXPERIMENT USING SUCCESSOR VARIETY.” *AN EXPERIMENT USING SUCCESSOR VARIETY*.

Anon. 2020. “Amharic Light Stemmer.” *Amharic Light Stemmer* (ii):1–10.

Anon. n.d. “Information Retrieval_ CHAPTER 8_ STEMMING ALGORITHMS.”

Argaw, Atelach Alemu, and Lars Asker. 2007. “An Amharic Stemmer : Reducing Words to Their Citation Forms.” (June):104–10.

Bade, Girma Yohannis, and Hussien Seid. 2018. “Development of Longest-Match Based Stemmer for Texts of Wolaita Language.” 4(3):79–83.

Bellovin, Steven M., and Eric K. Rescorla. 2005. “Deploying a New Hash Algorithm.” 1–10.

Demilie, Wubetu Barud. 2019. “Parts of Speech Tagger for Awngi Language.” 9(9).

Ehret, U., H. V Gupta, M. Sivapalan, S. V Weijs, S. J. Schymanski, G. Blöschl, A. N. Gelfan, and C. Harman. 2014. “Advancing Catchment Hydrology to Deal with Predictions under Change.” 649–71.

Ferro, Nicola. 2002. “University of Padua at CLEF 2002 : Experiments to Evaluate a Statistical Stemming Algorithm.”

Haroon, Muhammad. 2018. “Comparative Analysis of Stemming Algorithms for Web Text Mining.” (September):20–25.

Ismailov, A., M. M. Abdul Jalil, Z. Abdullah, and N. H. Abd Rahim. 2016. “A Comparative Study of Stemming Algorithms for Use with the Uzbek Language.” *A Comparative Study of Stemming Algorithms for Use with the Uzbek Language* (December).

Kraaij, Wessel. n.d. “Porter ’ s Stemming Algorithm for Dutch.” 167–80.

Otair, Mohammed Abdallh. 2016. “COMPARATIVE ANALYSIS OF ARABIC.” (May 2013).

Samuel, Jonathan, Solomon Teferra, Jonathan Samuel, Solomon Teferra, Jonathan Samuel, and Solomon Teferra. 2018. “Designing A Rule Based Stemming Algorithm for Kambaata Language Text.” (9):41–54.

Sever, Hayri, and Yıltan Bitirim. n.d. “FindStem : Analysis and Evaluation of a Turkish.” 238–39.

Sousa, Filipe J., and Luis M. De Castro. n.d. “Of the Significance of Business Relationships *.”

Stein, Benno, and Martin Potthast. n.d. “Putting Successor Variety Stemming to Work.”

Synthesizer, Morphological, and Abebe Abeshu. 2013. “Analysis of Rule Based Approach for Afan Oromo Automatic.” 7522(4):94–97.

Tesfaye, Debela. 2010. “ADDIS ABABA UNIVERSITY FACULTY OF INFORMATICS DEPARTMENT OF INFORMATION SCIENCE Designing a Stemmer for Afaan Oromo Text : A Hybrid Approach SCHOOL OF GRADUTE STUDIES FACULTY OF INFORMATICS.”

Tesfaye, Debela. 2011. “A Rule-Based Afan Oromo Grammar Checker.” 2(8):126–30.

Tesfaye, Debela. n.d. “Designing a Rule Based Stemmer for Afaan Oromo Text.” (1):1–11.