Intelligent Information Management, **2013, 5, 136-139**
http://dx.doi.org/10.4236/iim.2013.54014 Published Online July 2013 (http://www.scirp.org/journal/iim)

A Hybrid Algorithm for Stemming of Nepali Text

Chiranjibi Sitaula
Central Department of Computer Science and Information Technology, Tribhuvan University, Kathmandu, Nepal
Email: candsbro@gmail.com

Received April 25, 2013; revised May 26, 2013; accepted June 15, 2013

ABSTRACT

In this paper, a new context free stemmer is proposed which consists of the combination of traditional rule based system with string similarity approach. This algorithm can be called as hybrid algorithm. It is language dependent algorithm. Context free stemmer means that stemmer which stems the word that is not based on the context *i.e.*, for every context such rule is applied. After stripping the words using traditional context free rule based approach, it may over stem or under stem the inflected words which are overcome by applying string similarity function of dynamic programming. For measuring the string similarity function, edit distance is used. The stripped inflected word is compared with the words stored in a text database available. That word having minimum distance is taken as the substitution of the stripped inflected word which leads to the stem of it. The concept of traditional rule based system and corpus based approach is heavily used in this approach. This algorithm is tested for Nepali Language which is based on Devanagari Script. The approach has given better result in comparison to traditional rule based system particularly for Nepali Language only. The total accuracy of this hybrid algorithm is 70.10% whereas the total accuracy of traditional rule based system is 68.43%.

Keywords: String Similarity; Information Retrieval; Text Mining; Natural Language Processing; Dynamic Programming

1. Introduction

Stemming means finding the root or stem from the given inflected word. It is used in Natural Language processing, Information Retrieval, Text Mining etc. Mostly, stemming is used to improve the performance for NLP (Natural Language Processing). For example, if the word such as "उपरथिहरु" is used for NLP. Searching with this long string may degrade the performance, but if the stemming is done with this word *i.e.* रथि. Obviously, the performance is increased because we don't need to search other unnecessary words.

Apart from the Natural Language Processing Task, the stemming plays a very important role in text mining task of computer science. In the process of stemming normally the input tokens are given the core engine which strips the inflected word leading to proper root which is used for better searching in search engine (**Figure 1**).

For stemming purpose, different algorithms are available in text mining purpose. They include rule based, machine learning based and statistical based algorithms.

The input and output of the data are given in **Table 1** for the demonstration. The required output is given into second column whereas the word to be tested is given in the first column.

2. Literature Review

Several works have been performed in the field of stemming including German, Spanish, Indian and etc. Talking about Indian stemming which is more similar to Nepali

Table 1. Mapping of inflected word and stem/root.

Inflected word	Root
उपरथिहरु	रथि
समाचार	आचार
संवेदना	वेदना
औपन्यासिक	उपन्यास
अत्यावश्यक	आवश्यक

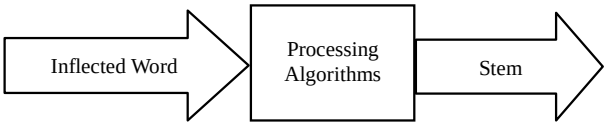

Figure 1. Processing for Stemmer.

stemming, have performed stemming work in their own different local languages like Tamil, Punjabi, Bengali, Gujarati, Hindi, Marathi etc. Similarly, Arabians have also performed such operation in their local language. Although many algorithms exists, they are mainly focused on their native language only.

[1] performed stemming approach in Arabic text. They used five methods. Of which four of these were positional letter ranking approach and fifth was traditional rule based system and found that rule based system performed well when combining with correction algorithm. Similarly, [2] used machine learning approach for stemming approach and it performed high accuracy. For performing such stemming approach they used different classifier like Naïve Bayesian, Bayesian Network, OneR, ZeroR and J148 algorithms. [3] used light stemmer with heuristic and co-occurrence of information retrieval approach. For handling co-occurrence they used clustering approach. [4] used different existing stemming algorithms and comparisons is made. They include Lovins, Porter Stemmer etc. [5] discussed about different stemming algorithms for English Language and a hybrid algorithm is made for Gujarati incorporating different algorithms. Some performed stemming of the Arabic text using Hidden Markov model [6] which gave more accuracy. Similarly [7] stemmed the Hindi Texts using hybrid approach. In this approach, they used the combination of brute force and suffix stripping approach which tries to remove the problem of over-stemming and under-stemming. The very preliminary phase of Hindi was Light Stemmer which was done by [8]. It is just a set few rules with the help of which stemming are done. Similarly, [9] uses stemming algorithms for Punjabi words which are based on the algorithms defined by [10,11]. It exploits brute force approach with few stripping strategy. It just matches the patterns and displays the root from the database. It the searching word is found, its respective stem is retrieved otherwise it just stems the affix and gives the output. [12] used stemming approach for text classification in Arabic language. For preprocessing the texts, they used stemming and performed classification. Similarly, [12-17] has done research on their different languages like Bengali, Punjabi etc. [18] performed research of different stemming algorithms. [19] explains stemming algorithms for Arabic language using parallel corpus and [20] has explained different stemming algorithms. [21-23] also performed different stemming approach for their own language like Nepali and Turkish.

3. Proposed Model

For explaining the proposed model, following things are taken into consideration: prefix, suffix and root.

3.1. Suffix

Suffix means those words that come after the root or stem. Around 150 suffixes are taken into consideration.

3.2. Prefix

These are the words that are added to the front part of the Nepali words. Around 35 words are taken.

3.3. Roots

Around 700 complex stem words are taken into consideration for research activity.

3.4. Hybrid Methodology

This algorithm is based on classical rule based stemming algorithms like [8]. It exploits the features of string similarity function of dynamic programming [23]. The complete flow of algorithm is given in **Figure 2** below.

This algorithms in context free algorithm. *i.e.*, it doesn't care about the context of the word. It strips the words from the inflected word. For example, in this algorithm, after entering the inflected word the stripping operation is performed. Incremental stripping approach is

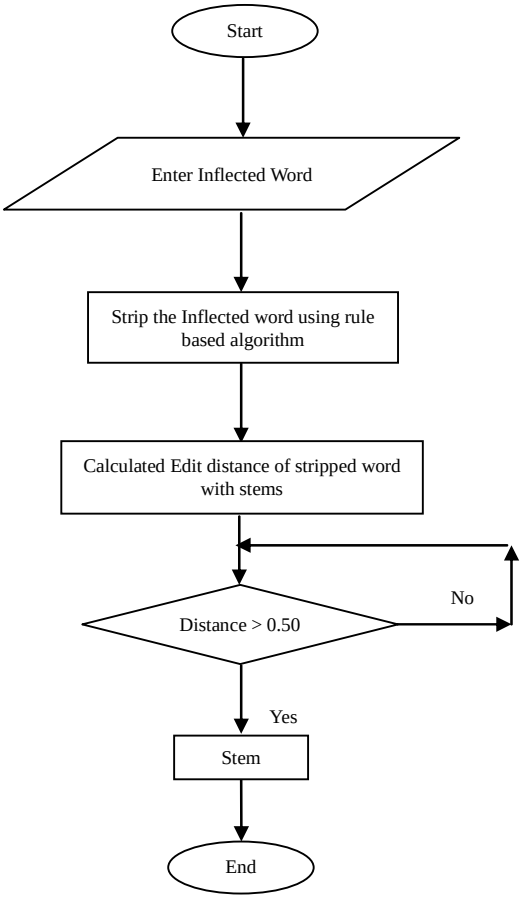

Figure 2. Flow of operation.

employed for suffix portion. But for prefix portion, it is not employed. For example, उपरथिहरुले = उप + रथि + हरु + ले. In this example, "हरु" and "ले" are two suffixes stripped through incremental approach and "उप" is prefix. Similarly, for prefix portion longest length stripping approach is followed. In this method, the longest size of the prefix is stripped although there may be presence of another affix. For example, अमान्छे = अ + मान्छे and अधिकरण = अधि + करण. In this example, "अ" and "अधि" both are prefix but priority is given to the prefix having longest size like "अधि" not "अ".

As 0.50 was found to be the best threshold in [24], it is taken as the best threshold. After stripping the word, the words are compared with the roots stored in database using string similarity function which has used dynamic programming approach for comparing.

The output obtained from this stop is stripped word but in some time, the word may be over stripped or under stripped. In order to compensate the over stripped or under stripped words, the concept of string similarity approach is exploited.

4. Evaluation and Output

For the evaluation purpose, around 1200 complex words are taken as test keywords. It was implemented under Visual Studio 2008. Programming language was C#. For measuring the performance, precision and recall are used.

The comparison of the hybrid algorithms with traditional rule based algorithm is made. The output is listed in **Table 2**.

Similarly, the output of the stemming using traditional rule based system is listed in **Table 3**.

5. Conclusion and Limitation

After performing the research on stemming of Nepali Keywords, following conclusions are made:

- The recall of rule based system was 68.43 and the recall of Enhanced system was 72.1.
- The over stripping and under stripping are recovered by Enhanced System.
- Its context free nature is not handled.
- Few rules are applied.
- Incremental stripping of prefix is not allowed.
- Longest length stripping is not applied in suffix portion although it is applied in prefix portion.
- It can be compared with many other algorithms.
- The less number of words stored in corpus leads to wrong output so more number of words are necessary in corpus.
- Different thresholds for measuring the distance can be used.
- Different similarity measures can be used and compared.

Table 2. Output of rule based system.

Group	No. of keywords	Recall	Average recall
1	400	75.25	
2	400	74.45	72.1
3	497	67.60	

Table 3. Output of modified algorithms.

Group	No. of keywords	Recall	Average recall
1	400	74.25	
2	400	70.45	68.43
3	497	60.60	

6. Acknowledgements

Thanks to Dr. Bipul Shyam Purkayastha from Assam University, India for providing me an implausible support. Similarly, special thanks go to Mr. Bikash Balam from Central Department of Computer Science and Information Technology, Tribhuvan University along with my students and colleagues for supporting me.

REFERENCES

[1] Y. AI-Nashashibi, D. D. Neagu and Y. Ali, "Stemming Techniques for Arabic Words: A Comparative Study," 2*nd International Conference on Computer Technology and Development* (*ICCTD*), 2010, pp. 270-276.

[2] H. Mohammad, B. Zuhair, C. Keely and M. David, "An Arabic Stemming Approach Using Machine Learning with Arabic Dialogue System," *ICGST AIML*-11 *Conference*, Dubai, April 2011, pp. 9-16.

[3] L. S. Leah, B. Lisa and C. E. Margaret, "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurance Analysis," *SIGIR*, ACM, 11-15 August 2002.

[4] L. S. Leah, B. Lisa and C. E. Margaret, "Conservatice Stemming for Search and Indexing," *ACM*, August 2005, pp. 15-19.

[5] S. Jikitsha and P. C. Bankim, "Stemming Techniques and Naïve Approach for Gujarati Stemmer," *International Conference in Recent Trends in Information Technology and Computer Science*, IJCA, 2012, pp. 9-11.

[6] A. F. Alajmi, E. M. Saad and M. H. Awadalla, "Hidden Makov Model Based Arabic Morphological Analyzer," *International Journal of Computer Engineering Research, IJCER*, Vol. 2, No. 2, 2011, pp. 28-33.

[7] M. Upendra and P. Chandra, "MAULIK: An Effective Stemmer for Hindi Lanuage," *International Journal of Computer Science and Engineering, IJCSE*, Vol. 4, No. 5, 2012, pp. 711-717.

[8] R. Ananthakrishnana and R. D. Durgesh, "A Light Stemmer for Hindi."

[9] K. Dinesh and R. Kumar, "Design and Development of

Stemmer for Pujabi," *International Journal of Computer Applications*, *IJCA*, Vol. 11, No. 12, 2010, pp. 18-23. doi:10.5120/1634-2196

[10] S. Llia, "Overview of Stemming Algorithms," Depaul University.

[11] F. B. William and F. J. Christopher, "Strength and Similarity of Affix Removal Stemming Algorithms," James Madison University and Virginia Tech.

[12] O. H. M. Ali and L. Ma Shi, "Stemming Algorithm to Classify Arabaic Documents," *Symposium on Progress in Information & Communication Technology*, 2009, pp. 111-115.

[13] A. James and K. Giridhar, "Stemming in the Language Modeling Framework," *SIGIR*, *ACM*, Toronto, 28 July-1 August 2003.

[14] A. Farag and N. Andreas, "N-Gram Conflation Approach for Arabic Text," *SIGIR*, *ACM*, Amsterdam, 7 July 2007.

[15] K. Dinesh and R. Prince, "Stemming of Punjabi Words by Using Brute Force Technique," *International Journal of Engineering Science and Technology*, *IJEST*, Vol. 3, No. 2, 2011.

[16] D. Sajib and N. Vincent, "Unsupervised Morphological Parsing of Bengali," Lang Resource & Evaluation, Springer, 2007.

[17] R. Monica, M. Scott and Y. Yiming, "Unsuperised Learning of Arabic Stemming Using a Parallel Corpus," *Proceeding of the* 41*st Annual Meeting of the Association for Computation Linguistics*, July 2003, pp. 301-398.

[18] N. S. Giridhar, K. V. Prema and N. V. Subba Reddy, "A Prospective Study of Stemming Algorithms for Web Text Mining," *Ganapt University Journal of Engineering & Technology*, Vol. 1, 2011, pp. 28-34.

[19] K. Chouvalit and B. Veera, "Inverted Lists String Matching Algorithms," *International Journal of Computer Theory and Engineering*, Vol. 2, No. 3, 2010, pp. 352-357.

[20] K. Koudas, S. Sunita and S. Divesh, "Record Linkage: Similarity Measures and Algorithms."

[21] J. Ms. Anjali, "A Comparative Study of Stemming Algorithms," *IJCTA*, Vol. 2, No. 6, 2011, pp. 1930-1938.

[22] B. Bal Krishna and S. Prajol, "A Morphological Analyzer and a Stemmer for Nepali," Madan Puraskar Pustakalaya, Working Papers 2004-2007.

[23] F. Cuna Ekmekcioglu, L. F. Michael and W. Peter, "Stemming and N-Gram Matching for Term Conflation in Turkish Texts," *Information Research*, Vol. 2, 1996.

[24] C. Sitaula, "Semantic Text Clustering Using Enhanced Vector Space Model Using Nepali Language," *GESJ*, Vol. 36, No. 4, 2012, pp. 41-46.

