MAULIK: An Effective Stemmer for Hindi Language

Upendra Mishra
M.TECH: Department of CSE
Lovely Professional University
Jalandhar, INDIA
upendra.mishra13@gmail.com

Chandra Prakash
Assistant Professor: Department of CSE
Lovely Professional University
Jalandhar, INDIA
cpiiitm2010@gmail.com

***Abstract*— In this paper, a new stemmer has been proposed named as "*Maulik*" for Hindi Language. This stemmer is purely based on Devanagari script and it uses the Hybrid approach (combination of brute force and suffix removal approach). Stemming can be used to improve the effectiveness of information retrieval. The proposed stemmer is both computationally inexpensive and domain independent. The results are favorable and indicate that the proposed stemmer can be used effectively in Information Retrieval systems. This stemmer also reduces the problem of over-stemming and under-stemming which was found in A Light weight Stemmer for Hindi.**

Keywords- Hindi Stemmer, Maulik, Brute Force Algorithm, Suffix Striping, Morphology, Information Retrieval .

I. INTRODUCTION

In Information retrieval techniques, stemming act as an important tool to deal with the vocabulary mismatch problem, in which query words do not match document words. Stemming is the process of reducing a derived word into its stem word. Numbers of stemmers have been developed for different languages, which takes on to reduce a word to its root form, thereby reducing the size of index.

Stemming comes under Natural Language Processing techniques. NLP is a field of Computer Science and linguistics concerned with the interactions between computers and human (natural) languages. It is branch of Artificial Intelligence.

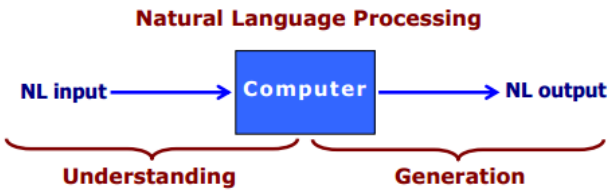

Figure 1: Natural Language Processing

Natural language processing encompasses anything a computer needs to understand natural language that is typed or spoken and it also generate the natural language. The task of Natural language understanding (NLU) [1] is understanding and reasoning while the input is a natural language. Here we can ignore the natural language generation issues. Natural language generation (NLG) is a subfield of NLP and it is also referred to text generation.

In theory, natural language processing is a very attractive method of Human computer interaction. Stemming is used in Information Retrieval systems where query words do not match vocabulary. For example, when a user enter a query word भारतीय (Bhartiya). If the exact word is not in the present in the vocabulary of the document then it may cause unreliable result. With the help of a stemmer, one can reduce the derived word into its root or stem word. In the previous case भारत (Bharat), is the stem or root word. Some of the Hindi Language stemming examples are shown in Table 1. Thus, using a stemmer improves recall, i.e., the number of documents retrieved in

response to a query. Also, since many terms are mapped to one, stemming serves to decrease the size of the index files in the IR system.

Table 1: HINDI STEMMING EXAMPLES

Input	Output
लाभदायक	लाभ
भारतीय	भारत
खेलना	खेल
लिखाई	लेख

The two main advantages of stemming algorithms are space efficiency and retrieval generality. The size of the inverted file can be reduced dramatically because many different words are indexed under the same stem and require only a single entry in the inverted file. Also, the generality is enhanced as query terms no longer have to match the text exactly [2]. The process of stemming is termed as conflation. The term conflation is used to denote the act of mapping variants of a word to a stem word. This research paper advocates an enhancement over previous stemmer which was based on suffix removal approach. In this paper a stemmer, *Maulik* is proposed that uses a Hybrid approach that combines suffix removal and brute force technique.

Details of this system are described in the remainder of this paper. The paper is organized as follows: Section (II) covers Approaches for Stemmer system; in section (III) methods for Stemming for Hindi language is discussed. In Section IV, Methodology for HINDI Stemmer, *Maulik* has being proposed. Results of evaluating the developed techniques are presented in section V. Finally, conclusions with limitation and future work are summarized.

II. PREVIOUS APPROACHES FOR STEMMING

In the areas of computational linguistics and information retrieval, Researchers find stemming a desirable step. In automated morphological analysis, the root of a word may be of less immediate interest than its suffixes, which can be used as clues to grammatical structure.

Stemming is not a new idea. Stemmer techniques has been developed since 1968. The first paper on the stemmer was written by Julie Beth Lovins [3]. Several researchers continued investigating various approaches to this problem was proposed in seventies and eighties. A later stemmer was written by Martin Porter and was published in the July 1980 issue of the journal Program [3]. As compared to English, a limited amount of research has been proposed in Hindi Stemming. A Light weight stemmer for Hindi was proposed by Durgesh D Rao & Ananthakrishnan Ramanathan[4] based on suffix striping approach .Yet stemmer for other languages like Nepali, Bengali and Punjabi ,Arabic has been proposed but research is mainly focused on English language. A statistical Hindi stemmer [5] was developed and used for evaluating the performance of the Hindi information retrieval system. Similar work has been done by Dasgupta and Vincent Ng [7] for Bengali Morphological analyzer.

There are different approaches used for stemming. An *Affix –Removal Approach* [8] is one of the simples techniques that uses a list of frequent affixes (prefixes and suffixes) to convert words to their root .There are two categories of affix removal stemmers, these are: the longest-match and Simple-removal. Another type of approach propsed is *N-gram technique* [9]. It does not produce an actual stem of a word; rather it computes how close two words are similar. It can be extended to compute, through a matrix, the similarity between every pair of terms in a document. Once such a matrix is obtained it can be used to calculate clusters. A shortcoming of such an approach is that for every word to be "stemmed" there must be procedure to identify its corresponding cluster and thus its representative.

In *Suffix Striping approach*, a pre-processing step is required in a number of natural language processing applications such as information retrieval, text summarization, document clustering, and word sense disambiguation. One of the quite widely used tool for this processing is stemmer which uses a suffix list to remove suffixes from words [5]. *Brute Force Approach* [10] approach employs a lookup table which contains relations between root words and inflected words. Whenever inflected word is entered then brute force searching is performed, in which it checks whether the inflected word exists in the table or not.

Another major stemming approach is *stochastic algorithms*, which uses probability to identify the root form of a word. These algorithms are trained to develop a probabilistic model. This model is form of complex linguistic rules. Stemming is performed by inputting an inflected form to the trained model and having the model produce the root form according to its internal rule set, which again is similar to suffix stripping and lemmatization.

In this research a new approach known as *Hybrid Approach* has been proposed for the stemming for Hindi words. This approach is combination of suffix stripping technique and brute force technique together[7]

III. STEMMING FOR HINDI

Hindi is the third most widely spoken language after English and Mandarin. It is spoken by approximately 600 million people all over the world [12]. Stemming from Sanskrit and being a part of the Indo-Aryan group of languages, Hindi draws on a host of different languages such as Persian, Turkish, Arabic and English. Hindi first started to be used in writing during the 4th century AD. It was originally written with the Brahmi script but since the 11th century AD it has been written with the devanagari script. Devanagari script is a straightforward and words exactly the way they are pronounced, unlike in English. Much of the vocabulary of Hindi comes from Sanskrit; through Hindi also has a special relationship with Urdu. Hindi is normally spoken using a combination of 52 sounds - 12 vowels, 35 consonants, nasalisation and a kind of aspiration. Here is the list of vowels, consonants and the other symbols that are used in Devanagari.

A. Vowels (स्वर)

Devanagari script has 10 vowels and two modifiers which are shown below in the table 2. The symbols shown below the alphabets are known as "matra" symbols. Matra symbols are used when consonants and vowels are to be written together.

Table 2: LIST OF VOWLES

अ	आ	इ	ई	उ	ऊ
	ा	ि	ी	ु	ू
a	aa/A	e/i	ee/ii	u	oo/uu
ए	ऐ	ओ	औ	अं	अः
े	ै	ो	ौ		
e	ai	o	ou	aM	aH

B. Consonants (व्यंजन)

There are 35 consonants in Hindi, and they are divided into three Groups that are Mutes, Semi vowels and Sibilants. First 25 alphabets are comes under mutes consonants, next 4 alphabets are comes under semi vowels and remaining are comes under Sibilants. There is a list of characters that are used to make a Hindi word. Without understanding and studying these words we could not speak and read Hindi language.

Table 3 : LIST OF CONSONANTS

क ka	ख kha	ग ga	घ gha	ङ ṅa
च ca	छ cha	ज ja	झ jha	ञ ña
ट ta	ठ tha	ड da	ढ dha	ण ṇa
त ta	थ tha	द da	ध dha	न na
प pa	फ pha	ब ba	भ bha	म ma
य ya	र ra	ल la	व va	श śa
ष ṣa	स sa	ह ha	क्ष ksa	त्र tra
ज्ञ jña				

C. Hindi Suffixes (प्रत्यय)

Suffix is the affix that is attached after the stem of a word. In this way, a new word is formed. There are two types of suffix one is primary suffix and another one is Nominal suffix (Figure 2). Primary suffix comes after the root of a verb. These suffixes are mostly present with the inflected words. Table 4 shows the different suffixes used in Hindi.

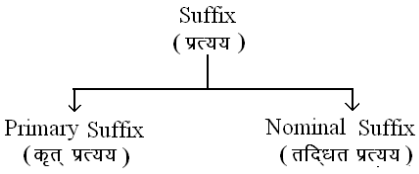

Figure 2: Suffixes in Hindi

Table 4: LIST OF SUFFIXES IN HINDI LANGUAGE

आ	इ	ई	नी	वान	ता	मान	पा	इया	वासी
तर	आनी	कार	आना	ईळा	ओं	ड़ा	आहट	गर	याँ

IV. PROPOSED METHODOLOGY FOR HINDI STEMMER, *MAULIK*

In this paper a new Stemmer for Hindi named "Maulik" has been proposed that uses a Hybrid Approach (combination of Brute force approach and Suffix Striping Approach). It is purely based on Devanagari script.

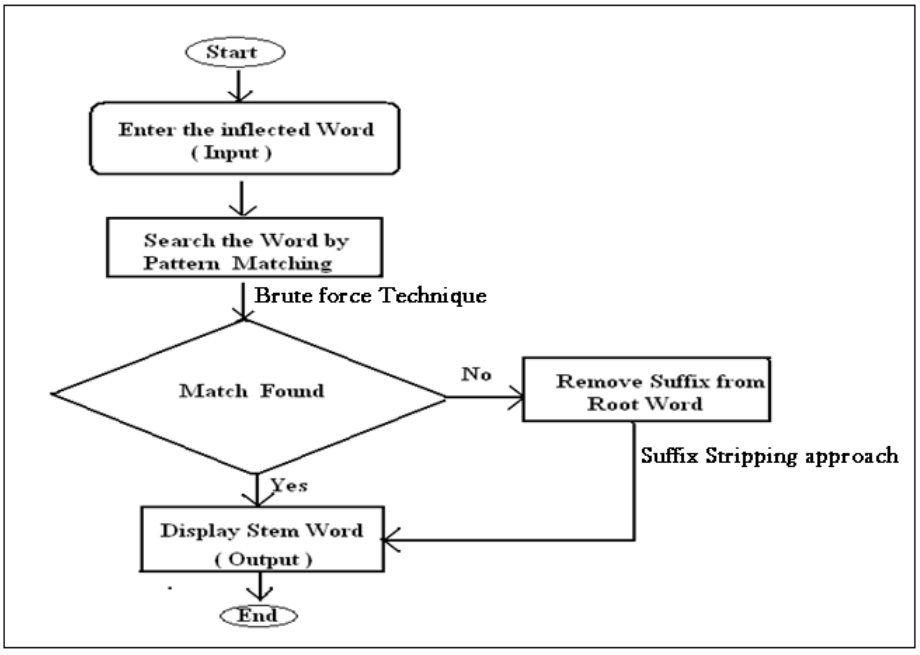

Figure 3: Methodology for Maulik , Hindi Stemmer

Brute force search[11] is also known as exhaustive search also known as generate and test this is a systematical approach which search for all the possible solution in the data. This approach employs a lookup table which contains relations between root words and inflected words. In this approach we can create store a large number of inflected words as well as their corresponding root word also. Whenever user gives the input then brute force searching is performed and it checks that whether the inflected words exists in the database or not, if the word is found then it simply gives its corresponding root word. If the word is not found in the database then it will go for using suffix stripping approach to handle these words. Suffix striping is basically a rule based

approach in which certain predefined rules are defined. On the basis of those rule suffixes are removed from the inflected word.

The processing of the inflected word for stemming is done in three steps. The steps are shown below.

a) Input Unit:

The inflected word is entered as an input. Here "भारतीय" is given as an Input.

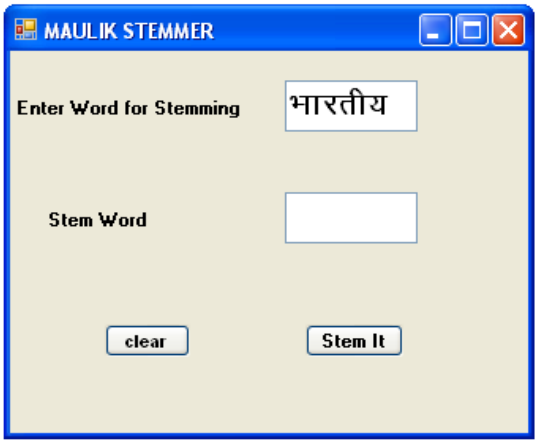

Figure 4: Input Unit

b) Processing Unit

The word is searched by the technique of brute force searching. Brute force search try to match the input word with the number of words present in the database. If the matching word is found then it will give the output as a stem word. If the match not found then it look for the alternate approach named suffixes stripping approach. It removes the matching suffix from the end of the word.

c) Output Unit

In Output Unit, The result comes after the processing of word. The result after processing is "भारत".

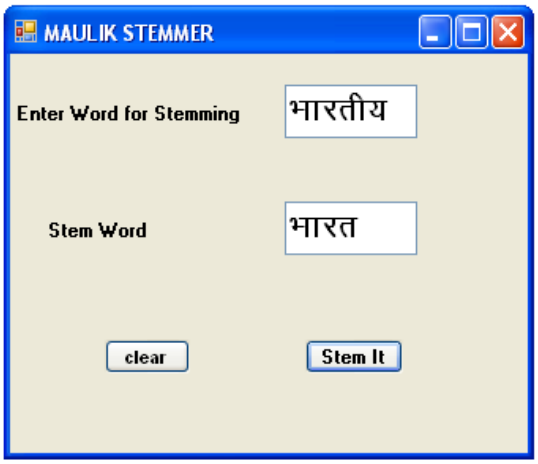

Figure 5: Output Unit

V. EVALUATION

The measure of the system's quality is the usual notion of the information-retrieval criteria. Parameters that are used for evaluating the proposed stemmer are Accuracy of stemmed word; Effectiveness of stemmer and Performance of stemmer.

Accuracy is directly proportional to the size of the database and the Rules that has been created for removing the suffix.15000 words are considered in lookup table therefore it reduces the chances of going to the second option of suffix stripping. Bigger database results in higher Accuracy.

Effectiveness is another important parameter for evaluating the stemmer. Effectiveness depends on the behavior of the stemmer. In this stemmer we have remove more than 50 suffixes for improving the effectiveness. Behavior means what stemmer will do whenever an abnormal condition occur Abnormal condition means whenever somebody tries to enter a word which does not exists. This stemmer is very effective if any user tries to enter a word that does not exists it just shows that word in the output box. Overstemming and under-stemming problem are also controlled in the stemmer. These errors are more whenever we are using suffix removal technique. By using brute force technique we have controlled these errors. If the inflected word exists in the database then there is no chance of these errors to occur.

Performance is directly proportional to the accuracy. For calculating the performance we have taken the 7 groups. Each group consists of 10 persons and each group tested the different number of inflected words. as the table shows that 4th group tested maximum number of words and 5th group test the minimum number of words.

Here first of all we can calculate the accuracy for each group and then calculate the accuracy of our stemmer by calculating the average accuracy of the groups. Average accuracy of our stemmer is 91.59%.

$$\text{ACCURACY}(\%) = \frac{\text{Acccurate words after Stemming}}{\text{Number of Inflected words entered}} * 100$$

$$\text{AVERAGE ACCURACY}(\%) = \frac{\text{Sum of All Accuracy}(\%)}{\text{Number Of Groups}}$$

Table 5: EVALUATION RESULTS

S.No	Number of Groups For Testing	Number of Inflected words Entered	Accurate words after Stemming	Accuracy
01	1st Group	320	294	91.87%
02	2nd Group	270	250	92.59%
03	3rd Group	420	380	90.47%
04	4th Group	450	420	93.33%
05	5th Group	180	165	91.67%
06	6th Group	375	345	92.00%
07	7th Group	250	223	89.20%

VI. CONCLUSION

This stemmer is purely based on Devanagari script and its gives the accuracy of 91.59%. This stemmer reduces the problem of understemming and over stemming. In future we can also create the stemmer by merging other techniques. The Limitation of the system is lack of human interference in the final result. As a future scope we can take the feed of user to make it more accurate. Also we would like to enhance our stemmer by including some more data in pure database and also by using extra rules for suffix stripping approach.

REFERENCES

[1] http://www.cnlp.org/publications/03nlp.lis.encyclopedia.pdf

[2] Trends in Information Retrieval by K.R. Chowdhary, Associate Professor, Dept. of Computer Sc. & Engg. Dr. V.S. Bansal, Professor Emeritus, Dept. of Electrical Engg.M.B.M. Engg. College, J.N.V. University, Jodhpur.

[3] Julie Beth Lovins, (1968)"Development of a Stemming Algorithm*"Mechanical Translation and Computational Linguistics, Vol No.11, Issue No.1, pp 22-31.

[4] M.F Porter (1980)" An algorithm for suffix stripping" Published in Program, Vol No.14, Issue No.3, pp 130-137,URL:http://www.cs.odu.edu/~jbollen/IR04/readings/readings5.pdf.

[5] Ananthakrishnan Ramanathan and Durgesh D Rao "A Lightweight Stemmer for Hindi" In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL), on Computatinal Linguistics for South Asian Languages (Budapest, Apr.) Workshop,pp 42-48.

[6] A. Chen and F. C. Gey, "Generating statistical Hindi stemmers from parallel texts," ACM Trans. Asian Language Inform. Process. Vol. 2(3), 2003.

[7] Sajib Dasgupta, Vincent Ng, "Unsupervised morphological parsing of Bengali," 2007

[8] Abduelbaset M. GOoweder, Husien A. Alhammi , Tarik Rashed, and Abdulsalam Musrati " A Hybrid Method for Stemming Arabic Text " Journal of computer Science,URL: http://eref.uqu.edu.sa/files/eref2/folder6/f181.pdf

[9] James Mayfield and Paul McNamee "Single N-gram Stemming" SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada,pp 415-416

[10] Prince Rana and Dinesh Kumar "Design and Development of a Stemmer for Punjabi" International Journal of Computer Applications (0975 – 8887)Volume 11– No.12, December 2010.

[11] Ababneh Mohammad, Oqeili Saleh and Rawan A. Abdeen(2006)" Occurrences Algorithm for String Searching Based on Brute-force Algorithm" Jordan Journal of Computer Science Vol No.2,Issue No.1,pp 82-85.

[12] http://www.outsourcingtranslation.com/resources/history/hindi-language.php

[13] http://www.hindilearner.com/hindi_alphabet.html

[14] R.S. McGregor. 1977. Outline of Hindi Grammar. Oxford University Press, Delhi, India.

[15] Jiaul H. Paik and Swapan K. Parui (2008) "A Simple Stemmer for Inflectional Languages" Journal of Documentation, Vol No.61 Issue No.4, pp. 476 – 496.

[16] Miguel E. Ruiz and Bharath Dandala (2010) "Evaluating Stemmers and Retrieval Fusion Approaches for Hindi: UNT at FIRE 2010" URL:http:// www.isical.ac.in/ ~fire/paper_2010 /MiguelRuiz-unt-fire-2010.pdf.

[17] Mudassar M. Majgaonker and Tanveer J Siddiqui(2010) "Discovering suffixes: A Case Study for Marathi Language" International Journal on Computer Science and Engineering Vol. 02, No. 08, 2010, 2716-2720.

AUTHORS PROFILE

Mr. Upendra Mishra is currently pursuing Masters of Technology from Lovely Professional University (LPU), Phagwara, India. He did his Bachelors of Technology (BTech) from College of Engineering & Management Kapurthala, Punjab. His Research area of interest includes Natural Language Processing and Machine Learning.

Mr. Chandra Prakash is currently working at the rank of Assistant Professor in Computer Science Department of Lovely Professional University (LPU) , Punjab, India. He did his Integrated Masters (BTech and MTech) in Information Technology from Indian Institute of Information Technology and Management Gwalior in 2010. His areas of research are Data Mining, Speech Recognition, Text summarization, Machine learning, Human-Computer Interfaces and Biometrics. He also dabble a lot with Web based and Application Software. He has published 5 papers in various international and national journals/conferences. He is the winner of Lord of the Code Scholarship Contest organized by KReSIT, IIT Bombay and Red Hat, besides numerous other awards. He had worked with Center for Artificial intelligent and robotics, DRDO, Bangalore, India as trainee.