ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Proposed Stemming Algorithm for Hindi Information Retrieval

Anshu Sharma[1], Rakesh Kumar[2], Vibhakar Mansotra[3]

M. Tech Student, Dept. of Computer Science & I.T., University of Jammu, Jammu, J&K, India[1]

Research Scholar, Dept. of Computer Science & I.T., University of Jammu, Jammu, J&K, India [2]

Professor, Dept. of Computer Science & I.T., University of Jammu, Jammu, J&K, India [3]

ABSTRACT: Stemming maps morphologically related words to a common stem or root word by removing their suffixes or prefixes. It is basically an operation that reduces inflected word to its root form, but it is not necessary that stemming always provide us relevant and meaningful root word. For example word नदियों(nadion) in Hindi has a suffix ियों, if we remove this suffix from the word then it becomes नद (nadh), which is not a proper Hindi word. So, to overcome this problem, a stemming algorithm is proposed that uses Hybrid approach (combination of Brute Force Approach, Suffix Stripping Approach and Suffix Substitution).This stemming algorithm will be implemented for Hindi Noun words and further we extend this algorithm to other categories of Hindi words like verb, adverb, adjective . Results of this proposed algorithm will be implemented on Web based Interface.

KEYWORDS: Stemming, Brute Force Approach, Suffix Stripping Approach, Suffix Substitution Approach, Hindi Stemmer, Stop Words

I. INTRODUCTION

Stemming[1] is a process that reduces the morphological similar words to their root form. For example Hindi words like कविताओ, कविताएं, कविता all derived from a common root word कविता.Stemming maps all these words (कविताओ, कविताएं, कविता) to their root form (कविता). The basic process of stemming is shown in fig 1.

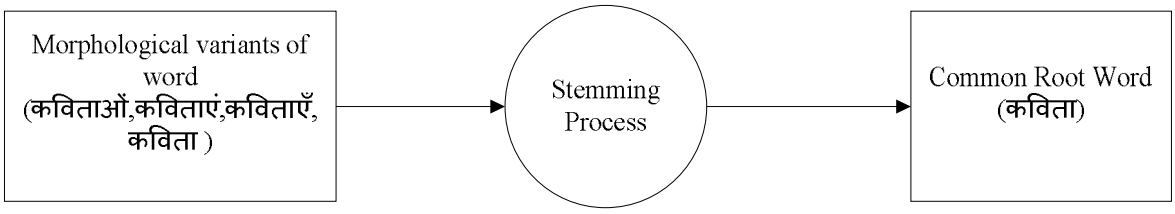

Fig.1 Basic Process of Stemming

In this paper, a stemming algorithm is proposed that remove stop words from the given query and perform stemming on words to get relevant results. Both stop words removal and stemming are most commonly text operations in Information Retrieval System. Stop word elimination removes grammatical or functional words while stemming reduces words to their common root words. When perform stemming, one should take care about over stemming and under stemming. Under stemming occurs when two related words are not reduced to the same stem . Over stemming occurs when two unrelated words are reduced to the same stem, thereby cause match between query and irrelevant documents. Example below shows stemming of some Noun words:

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Inflected Word	Output
पक्षियों(pakshion)	पक्षी(pakshi)
मालाओं(malaon)	माला(mala)
सेवाएं(sevaian)	सेवा(seva)
कहानियाँ(kahaniyan)	कहानी(kahani)

One main advantage of stemming is in indexing in IR because instead of storing all the words like " कविताओ, कविताएं,कविता ", only their root form " कविता " is stored . Stemming also improves Recall and Precision to a greater extent. The proposed Stemming algorithm that are presented by this paper uses the Hybrid approach (combination of Brute Force approach, Suffix stripping approach and Suffix substitution on the basis of rules). In this paper, we focused on Noun words only and this algorithm perform stemming on inflected Hindi noun words by removing suffixes from them, if after removing suffix again word is not present, then some rules are applied to make this word a proper Hindi word.

The paper is organized as follows: The next section discusses related work of stemming for Indian languages. Section3 looks at the various approaches to Stemming. Section 4 discusses Stemming for Hindi Language. Section 5 discusses proposed algorithm presented by this paper. Section 6 discusses Implementation of this proposed algorithm. Section 7 discusses Conclusion and at the last references are shown that are used in this paper.

II. RELATED WORK

Stemming is not a new idea. This has been developed since 1968.Lots of work had been done on stemming in English Language, but not much has been reported for Indian Languages. For Indian Languages, In [2] author had developed stemming algorithm for Hindi which was light weight and uses a hand crafted suffix list on a longest match basis. For testing, documents were chosen from varied domains such as Films, Health, Business, Sports and Politics. This stemmer calculates under-stemming and over-stemming error rates which are 4.68 % & 13.84% respectively.

In [3] author had proposed stemmer "MAULIK " for Hindi Language that is purely based on Devanagari Script. This Stemmer uses a Hybrid approach and gives accuracy of 91.59%.In [4] author uses the same technique but for Punjabi language and this stemmer gives average accuracy of 80.73%.In [5] author had developed Hindi rule Based Stemmer for Nouns. This stemmer has been tested on 100 news documents taken from popular Hindi newspapers. In this paper, author had used 16 Hindi suffixes. This stemmer gives 16.35% error due to absence of some noun suffixes. So, the author leaves improvement for future work .By adding more Hindi suffixes we can remove error to some extent.

In [6] author had also developed Suffix Stripping Based Verb Stemming for Hindi. This Hindi Verb Stemmer gives accuracy of 83.63% . In [7] author had developed unsupervised approach for the development of a stemmer(for Urdu &Marathi Language).Author had used two approaches for suffix stripping namely frequency based stripping and length based stripping. For Urdu Language frequency based suffix stripping gives accuracy of 85.36% whereas Length based suffix stripping algorithm gives accuracy of 79.76%. And for Marathi Language, accuracy is 63.5% in case of frequency based stripping and 82.5% in case of length based suffix stripping. In [8] author had developed rule based hindi lemmatizer. For this purpose, they had created 124 rules which used in such a way that initially suffix gets removed from the input word and if required, addition of new character takes place. For analysis purpose, system is being tested on 2500 words and it gives accuracy of 89.08%. In [9] Morphological Analyzer for Hindi is presented. The approach that was used in this paper is both rule based and corpus based. This Morphological Analyzer works for both Hindi words and sentences.In [10], author had developed Morphological Analyzer for Hindi which is a rule based implementation. This analyzer takes either Hindi sentence or word as input and generate its necessary features with its root words by analyzing it and works on both inflectional and derivational morphemes. When exception occurs using rule based approach, it also utilizes corpus. In [11], author had developed Hindi Morphological Analyzer and

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Generator. In this paper author performs the morphological analysis for Hindi Language using paradigm approach with Windows platform having GUI. This tool is built using Visual Basic at Front end and MS-Excess at back end.

III. STEMMING APPROACHES[4]

Different approaches which are used for Stemming are shown below:

A. *Affix Removal Approach:*

This is one of the simplest approach which is used for stemming. Affix Removal refers to either removing suffix or prefix from the input word which was entered by the user. This method is based on two principles, one is iteration and other is longest match. A set of rules for such a stemmer is as follows :If a word ends with suffix "एं" . Then remove "एं". An example for this rule is:

Hindi words like मालाएं,सेवाएं,आशाएं ends with suffix" एं", when we remove suffix "एं" from these words, then they reduces to their root form (माला,सेवा,आशा) which is the proper Hindi word. So, the user get these words easily from database. In the same way, one can make more rules for Hindi words.

B. *N-GRAM[12]***:**

This approach was given by Adamson and Boreham. There are frequently used subsets of n-grams, such as bigrams(digrams, bi-grams,di-grams), representing a sequence of two grams(two characters or two words or two syllables in a row, consecutively in the text)and trigrams(three characters in the text). In this approach, we first find unique pairs of words(di-grams) and then associate them on the basis of unique digrams they both possess. For example, the terms माला **(mala)** and मालाओं **(malaon)** can be broken into digrams as follows.

माला(mala) : मा ◌ाल ला.

Unique Digrams : मा ◌ाल ला.

मालाओं(malaon) : मा ◌ाल ला ◌ाओ ओं.

Unique Digrams : मा ◌ाल ला ◌ाओ ओं.

Thus , " माला " has three digrams, out of which all are unique, and " मालाओं " also has five digrams, out of which all are unique. These two words share three unique digrams," मा " " ◌ाल" " ला ".When we compute these unique digrams for the word pair , then similarity measure which is based on them is computed. This similarity measure is computed using Dice's coefficient[3], which is defined as: $S = 2C/A + B$**.**

C. *Suffix Stripping Approach[13]:*

This approach is based on removing suffixes from the word, if the word is not present in vocabulary. In this approach, suffix list is required with the help of which one can remove suffixes from the word. To make a decision , these approaches check the vocabulary for the existence of word. The non -existence of word may cause algorithm to try suffix stripping rules. It is possible in this case, that two or more suffix stripping rules apply to the same input term, so it will cause ambiguity in which rule to apply.

D. *Suffix Substitution Approach[13]:*

Suffix Stripping approach is an improvement upon Suffix Stripping Approach. For this, one needs to create substitution rule that replaces one suffix with another suffix.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

E. *Brute Force Approach[13]:*

Brute Force employs lookup table which contains relationship between root word and inflected word. Whenever user enters inflected word brute force searching is performed for checking whether inflected word is present in the table or not.

F. *Stochastic Algorithms*:

Stochastic Algorithm is another Stemming approach that uses probability to identify root form of word. To develop probabilistic model ,these algorithms are trained, so that one can perform Stemming by inputting inflected form of word to this model and generate root form of word according to its internal rule set.

G. *Hybrid Approach*:
Hybrid approach uses combination of two or more approaches.

The algorithm that are presented by this paper uses the Hybrid approach i.e. combination of Brute Force approach, Suffix Stripping approach and Suffix Substitution approach.

IV. STEMMING FOR HINDI LANGUAGE

After English and Mandarin, the most widely spoken language is Hindi. About 600 million people speak this language all over world[3]. Hindi is highly inflectional language and shows inflections in different form of words like noun, verb, adverb, adjective. The noun inflection is discussed below:

- **Noun Inflections**

According to[2], Hindi nouns shows inflection only for number and case. Number can be either singular or plural. In Hindi, cases are of two types-direct and oblique. For example -in word घोडा ,घोडे, ा shows singular number whereas े shows plural number. Similarly there are some gender rules for Hindi. In Hindi, words that ends with ी are considered as feminine whereas words that ends with ा are considered as masculine. For example words लडका, घोडा,गधा considered as masculine because they ends with ा. Similarly, words लडकी, पुत्री, घोडी, कटोरी are considered as feminine because they ends with ी.But there are some words that do not follow these rules. For example- word पानी is masculine but this word ends with ी. Similarly word माला is feminine although it ends with ा. So, it shows that Hindi is highly inflected language and require a deep study of word structure and its formation. In the same way, Hindi shows inflection for verbs, adverbs, adjectives. The objective of this research is to study inflections on Hindi Noun words and perform stemming on them.

V. PROPOSED ALGORITHM

In this paper, a Stemming algorithm for Hindi is proposed that uses Hybrid approach i.e. combination of Brute Force , Suffix Stripping and Suffix Substitution. Initially user will enters a query in the input field on a Web Based Interface and this algorithm removes stop words from this query and consider only those words that remains after removing stop words. After this, this stemmed query is searched in the database by using Brute force approach, if this is present in the database, then original word and related word is displayed on the interface. But if the stemmed query is not present in the database, then suffixes are removed from the noun words and again searched in the database, if the word after removing suffix is present ,then original word ,stemmed word, removed suffix and related word will be displayed on the web based interface in the form of table. If, in case again words are not present, then suffix substitution approach is applied wherever it is necessary. Rules are studied from previous literature[7] and some more rules are added to improve the accuracy of this proposed algorithm.

A. Stop Words

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Examples of most frequently stop words[13] which are used in this algorithm are shown below in Table 1.A total of 165 stop words were used in this algorithm, but this table shows only few of the stop words.

इत्यादि	द्वारा	इन्हें	इन्हों	हुई	मे	ने	पहले
इसमें	जितना	दुसरा	कितना	के	तिसे	सारा	नीचे
करने	किया	लिये	अपने	जिसे	होना	उनको	रहे
सबसे	होने	करते	बहुत	वर्ग	वहाँ	हैं	बाला
करता	उनकी	इसकी	सकता	जीधर			

Table1:Stop Words

B. Suffixes

Suffixes are used in this algorithm are shown below in Table 2:

े	ए	ी	ो	ई	ा	या
ौं	एे	ेें	एं	एँ	ोें	ओं
ीय	ना	ता	ती	जन	गण	िअो
यों	यां	त्व	कार	तया	वर्ग	ियों
ियां	िअोें	ियाँ	ियाँ			

Table2:Suffix List

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

C. Flow Chart

The flow chart of this proposed stemming algorithm is shown below in fig2 .

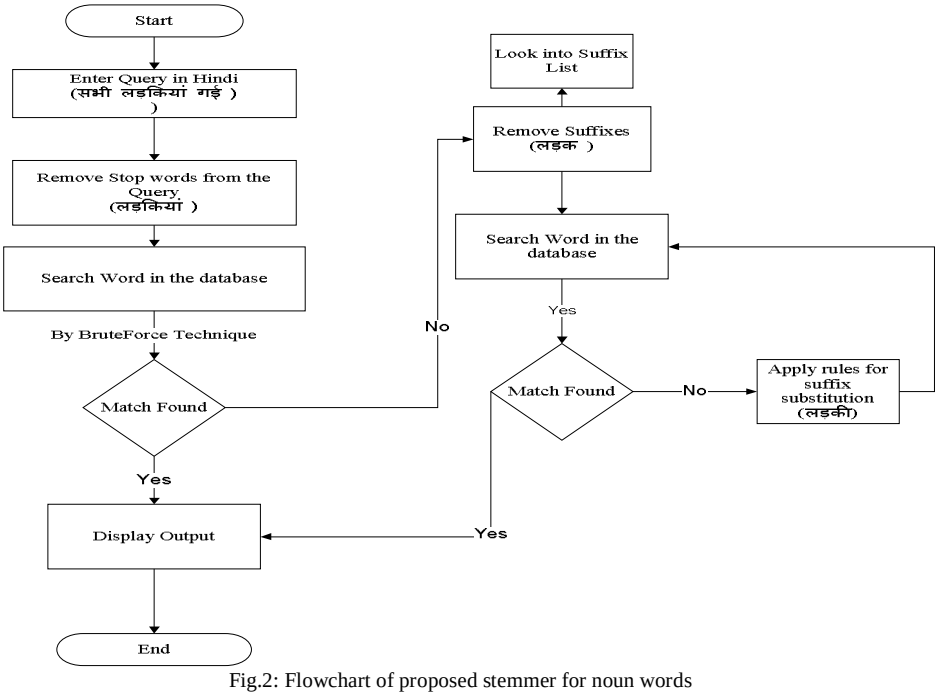

Fig.2: Flowchart of proposed stemmer for noun words

D. Rules for Suffix Substitution are:

- If word ends with any of the suffixes ियों or ियां or िअों or यां or यों.Then, remove these suffixes from the end of that word and add ी at the end of stemmed word. Search this result in the database, if the word is present then that word is returned as result.
- If word ends with suffix अों or ौं or एे, remove this suffix from the end of word and search into the database. If word is present ,then that word is returned as result.
- If word ends with any of the suffixes ों or े or ो. Then, remove these suffixes from the end of that word and add ा at the end of stemmed word. Search this result in the database, if the word is present, then that word is returned as result.
- If word ends with any of the suffixes ं or ी or एँ or एं or ा. Remove these suffixes from the end of that word. Search this result in the database, if the word is present, then that word is returned as result.
- If word ends with suffix ए. Then, remove this suffix from the end of that word and add आ at the end of stemmed word .Search this result in the database, if the word is present then that word is returned as result.
- If word ends with suffix ई or ना or ता or ती or जन or गण or तया or वर्ग or कार or त्व or ेंं or ीय , remove this suffix from the end of the word and search into the database. If word is present, then that word is returned as result.
- If word ends with suffix ियाँ. Then, remove this suffix from the end of that word and add या at the end of stemmed word . Search this result in the database, if the word is present then that word is returned as result.
-

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

VI. IMPLEMENTATION & EVALUATION

This proposed algorithm for stemming will improves the accuracy of existing stemmer for Hindi Language. User enters query in Hindi through external keyboard or Hindi Keypad which is created on Web Based Interface. We used a list of 165 stop words and 32 suffixes for this work. Both Stop words and suffixes are stored in an array for this algorithm. We select only those suffixes which are highly used with noun words and these are selected on the basis of study of different Hindi newspapers, different articles in Hindi which are available online. Results of this proposed stemming algorithm will be implemented on Web Based Interface which is developed in JQuery, HTML, CSS, JAVASCIPT. At the back end, PHP is used for database connectivity. For suffix substitution , some basic rules are made from the study of Hindi noun words. This stemming algorithm is evaluated on the basis of accuracy and performance which is defined as:

$$\text{Accuracy} = \frac{\text{Accurate results obtained using proposed stemming algorithm}}{\text{Total Number of inflected words entered}}$$

This proposed stemming algorithm is being tested on different documents taken from different Hindi newspapers , Magazines, journals which are available online. Accuracy of this proposed stemming algorithm is shown below in Table3.

Number of inflected words entered	Accurate results	Accuracy
500	461	92.2%

Table.3: Evaluation Results

VII. CONCLUSION

This Stemming algorithm will uses the Hybrid approach for performing Stemming of Hindi Noun words and gives accuracy of 92.2%. It includes only thirty two suffixes for Hindi Nouns and further we add more suffixes and more rules to improve the accuracy of this proposed stemming algorithm.

REFERENCES

1. https://en.wikipedia.org/wiki/Stemming.
2. Ramanathan and D. D. Rao, 'A Lightweight Stemmer for Hindi', Workshop on Computational Linguistics for South-Asian Languages, EACL,pp.42-48, 2003.
3. Upendra and Chandra Prakash, 'MAULIK:An Effective Stemmer for Hindi Language', International Journal on Computer Science and Engineering, Vol.4,pg.711-717,2012.
4. Dinesh and PrinceRana, 'Design and Development of a Stemmer for Punjabi' , International Journal of Computer Applications(0975-8887) Vol. 11-No. 12, pg.18-23,2010.
5. Vishal Gupta, 'Hindi Rule Based Stemmer for Nouns' , International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue 1, pp.62-65,2014.
6. Vishal Gupta, ' Suffix Stripping Based Verb Stemming for Hindi ', International Journal of Advanced Research in Computer Science and Software Engineering, Vol.4, Issue 1, pp.179-181, 2014.
7. Mohd. Shahid Husain, 'AN UNSUPERVISED APPROACH TO DEVELOP STEMMER', International Journal on Natural Language Computing (IJNLC), Vol. 1, No.2,pg.15-23,2012.
8. Snigdha, Mini, Nisheeth Joshi and Iti Mathur, 'DESIGN OF A RULE BASED HINDI LEMMATIZER", csit, pp. 67-74, 2013.
9. Mayuri, Pooja Khanna, 'Development of Morphological Analyzer for Hindi', International Journal of Computer Applications , Vol.95-No.17, pg.1-5 2014.
10. Ankita, Pramila, Shashi and Ajai, Hemant, 'Morphological Analyzer for Hindi-A Rule Based Implementation', International Journal of Advanced Computer Research, Vol-4 Number-1 Issue-14,pg.19-24,2014.
11. Vishal Goyal and Lehal, 'Hindi Morphological Analyzer and Generator',IEEE Computer Society Press,California,USA,pg.1156-1159,2008.
12. M. Hafer and S. Weiss 1974. "Word Segmentation by Letter Successor Varieties", Information Storage and Retrieval, 10, pg. 371-85.
13. Giridhar , Prema ,Reddy,"A Prospective study of Stemming Algorithm for Web Text Mining ",GANPAT UNIVERSITY JOURNAL OF ENGINEERING &TECHNOLOGY,Vol.-1,ISSUE-1,pg.28-34,2011.
14. http://www.ranks.nl/stopwords/hindi.