International Journal of Soft Computing 13 (5): 134-138, 2018
ISSN: 1816-9503

Extraction of Root Words Using Morphological Analyzer for Hindi Text

[1]Anjusha Pimpalshende and [2]A.R. Mahajan
[1]CMR College of Engineering, Hyderabad, India
[2]IT Government Polytechnic, Nagpur, India

Abstract: Stemming is a process of extracting words from text and turning them into index terms in an IR system. Stemmers are based upon the written and not the spoken form of the language. Word stemming is one of the most significant factors that affect the performance of a Natural Language Processing (NLP) application such as Information Retrieval (IR) system, part of speech tagging, machine translation system and syntactic parsing, text summarization. A stemmer converts morphologically identical words to root word without performing analysis of that term. Sometimes, if we remove suffix from the word then the word may not be 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).

Key words: Natural language processing, stemmer, suffix stripping, rule based, machine

INTRODUCTION

A stemming algorithm or stemmer, aims at obtaining the root of a word that is its morphological stem by clearing the affixes that carry grammatical or lexical information about the word. English is a weakly inflected language (you could ignore inflections and still get reasonable search results) but some other languages are highly inflected and need extra work in order to achieve high-quality search results. Stemming attempts to remove the differences between inflected forms of a word in order to reduce each word to it's root form. Stemming can be used to improve the efficacy of sentence retrieval. For many researchers Human Computer Interaction (HCI) is a significant field of interest. Natural Language Processing (NLP) is one of the core research areas that support HCI.

Hindi language stemmer is used to find the root words for inflected words. Sometimes if we remove suffix from the word then the word may not be 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). For instance भारतीय may be reduced to the root भारत. . Some examples of inflected words and their root word (Table 1).

Table 1: Inflected words and their root word

Input	Output
लाभदायक	लाभ
भारतीय	भारत
खेलना	खेल
लिखाई	लेख

MATERIALS AND METHODS

Ease of use: The two main advantages of stemming algorithms are space effectiveness and retrieval simplification. 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. The process of stemming is termed as fusion. The term fusion is used to denote the act of mapping variants of a word to a stem word. This proposed stemmer uses a hybrid approach that combines suffix removal and brute force technique and suffix substitution technique.

Early history: Stemming is not a new idea. Stemmer echniques have been developed, since, 1968. The first study on the stemmer was written by Julie Beth Lovins

Corresponding Author: Anjusha Pimpalshende, CMR College of Engineering, Hyderabad, India

(Mishra and Prakash, 2012). A later stemmer was written by Martin Porter and was published in the July 1980 issue of the journal program (Mishra and Prakash, 2012). 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 and Ananthakrishnan Ramanathan (Dinesh and Rana, 2010) based on suffix striping approach. A statistical Hindi stemmer (Gupta, 2014) was developed and used for evaluating the performance of the Hindi information retrieval system. Similar work has been done by Dasgupta and Vincent Ng (Husain, 2012) fors 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 (Rastogi and Khanna, 2014). 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 shortcom ing 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 (Gupta, 2014). Brute force approach (Agarwal *et al*., 2014) approach employs a look up 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.

RESULTS AND DISCUSSION

Stemming in Hindi: Hindi is normally spoken using a combination of 52 sounds-12 vowels, 35 consonants. Here, is the list of vowels, consonants and the other symbols that are used in Devanagari.

Vowels in Hindi:

अ आ इ ई उ ऊ ए ऐ ओ औ अं अः ऋ

Consonants in Hindi:

Suffix (प्रत्यय -Pratyaya) in Hindi: Suffix are very important elements of Hindi grammar. Many Hindi words can be created using suitable suffix. For example, the word "खिलौना" (Khilona-Toy) is a combination of noun "खेल" (Khel-Play/Game/Match) and suffix "औना" (aana) (Table 2).

Proposed Hindi stemmer: In this proposed research we are using hybrid approach, i.e., combination of Brute force, suffix stripping and suffix substitution. For the given document after segmentation and tokenization algorithm removes stop words from this text and consider only those words that remains after removing stop words. After this, words are searched in a word net available on iitb by using brute force approach if this is present in this, then original word and related word is displayed. But if the stemmed word is not present then search a file, file contains suffix list. After removal again search in word net. If in case again words are not present, then suffix substitution approach is applied wherever it is necessary. Rules are studied from previous literature (Husain, 2012) and some more rules are added to improve the accuracy of this proposed algorithm (Fig. 1).

Rules for suffix substitution are: If word ends with any of the suffixes ि◌य or ि◌यां or ि◌अ◌ो◌ं or यां or य. Then, remove these suffixes from the end of that word and add ◌ी at the end of stemmed word. Search this result in the word net if the word is present then that word is returned as result.

If word ends with suffix अ◌ो◌ं or ◌ौ◌ं or ए◌ं remove this suffix from the end of word and search into the word net. 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 wordnet if the word is present, then that word is returned as result.

Table 2: Examples of root words suffix

Suffix	Word root	Example word
अक्कड़	भूल	भुलक्कड़
(Akkad)	(bhul)	(Bhulakkad-one who forgets)
ऊ।	कमा	कमाऊ
(u)	(kamaa)	(Kamaau-one who earns)
इया	घट	घटिया
(iyaa)	(ghat)	(Ghatiya-bad)
वाला	पढ़	पढ़नेवाला
(vaala)	(pad)	(Padnevaala one who reads)
आऊ	टिक	टिकाऊ
(aau)	(tikau)	(Tikaau-Durable)
हार	होना	होनहार
(haar)	(hona)	(Honnaar-promising)
हार	लेन	लेनदार
(daar)	(len)	(Lendar-creditor)
आलू	दया	दयालु
(aalu)	(Daya-Mercy)	(Dayaalu-kind)
आन	उड़	उड़ान
(aan)	(ud)	(udaan)
आई	लिख	लिखाई
(aai)	(likh)	(Likhai-writing)
इया	सज	सजावट
(iyaa)	(saj)	(Sajaavat-decoration)
ई (	बोल	बोली
(ee)	(Bol)	(Boli-language)
आहट	घबरा	घबराट
(aahat)	(Ghabra)	(Ghabraat-nervousness)
आव	बह	बहाव
(aav)	(bah)	(Bahaav-flow)

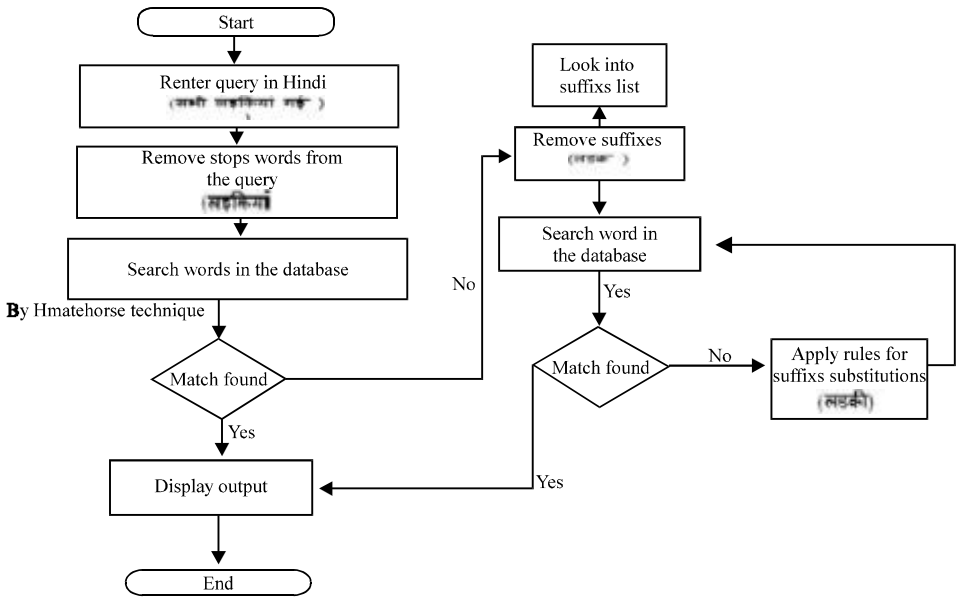

Fig. 1: Flowchart

If word ends with any of the suffixes ंं or ंी or ैं or ैं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.


Show 10 entries Search:

Before Removing Stop words	Before count	After Removing Stop words	After count	before Stremming	after Stremming
[illegible]	38	[illegible]	31	[illegible]	[illegible]
[illegible]	24	[illegible]	18	[illegible]	[illegible]

Fig. 2: Results of proposed stemming

Table 3: Proposed stemming

Number of inflected words entered	Accurate results	Accuracy
1000	920	92.2%

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 wordnet if the word is present then that word is returned as result.

If word ends with suffix ें or ना or ता or ती or जन or गण or तया or वग or कार or व or ंंंंंंंंंंीय remove this suffix from the end of the word and search into the wordnet. 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 wordnet if the word is present then that word is returned as result Alghorithm 1.

Algorithm 1: Input text document
Step 1: Input text document in Hindi
Step 2: perform segmentation and tokenization
Step 3: Remove stop words
Step 4: For each word check whether a word is ending with the given suffix if yes then remove it else perform brute force searching which checks whether the inflected words are present in a wordnet or not if it is present display it with root words
Step 5: If word is not present in Hindi word-net then perform its stemming procedure
Step 6: Match the suffix of the word with suffixes stored in a file. Remove suffix and search the stemmed word in Hindi word-net again
Step 7: If it is found then it is a valid word
Step 8: Current Hindi word is not a valid word apply suffix substitution
Step 9: End of procedure

Implementation and evaluation: This proposed algorithm for stemming will improves the accuracy of existing stemmer for Hindi language. User enters document in Hindi. We used a list of 165 stop words and 32 suffixes for this work. Both stop words and suffixes are stored in a file 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 text documents. Results of this proposed stemming algorithm will be implemented on java platform. For suffix substitution, some basic rules are made from the study of Hindi noun words. This stemming algorithm is evaluated on the basis of accuracyand performance which is defined as:

- Accuracy = Accurate results obtained using proposed stemming algorithm

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 in Table 3 and Fig. 3.

CONCLUSION

This stemming algorithm will uses the Hybrid approach for performing Stemming of Hindi words and gives accuracy of 92.2%. It includes only 32 suffixes for Hindi Nouns and further we add more suffixes and more rules to improve the accuracy of this proposed stemming algorithm.

REFERENCES

Agarwal, A., S.P. Singh, A. Kumar and H. Darbari, 2014. Morphological analyser for Hindi a rule based implementation. Intl. J. Adv. Comput. Res., 4: 19-25.

Dinesh and P. Rana, 2010. Design and development of a stemmer for Punjabi. Intl. J. Comput. Appl., 11: 18-23.

Gupta, V., 2014. Hindi rule based stemmer for nouns. Intl. J. Adv. Res. Comput. Sci. Software Eng., 4: 62-65.

Husain, M.S., 2012. An unsupervised approach to develop stemmer. Intl. J. Nat. Lang. Comput., 1: 15-23.

Mishra, U. and C. Prakash, 2012. MAULIK: An effective stemmer for Hindi language. Intl. J. Comput. Sci. Eng., 4: 711-717.

Rastogi, M. and P. Khanna, 2014. Development of morphological analyzer for Hindi. Intl. J. Comput. Appl., 95: 1-5.