Stop-Word Removal Algorithm and its Implementation for Sanskrit Language

Jaideepsinh K. Raulji
Lecturer, Ahmedabad
University, Ahmedabad,
Gujarat, India.
Research Scholar,
Dr. Babasaheb Ambedkar
Open University,
Ahmedabad, Gujarat, India

Jatinderkumar R. Saini, PhD
Professor & I/C Director,
Narmada College of
Computer Application,
Bharuch, Gujarat, India
Research Supervisor,
Dr. Babasaheb Ambedkar
Open University, Ahmedabad,
Gujarat, India

ABSTRACT

In the Information era, optimization of processes for Information Retrieval, Text Summarization, Text and Data Analytic systems becomes utmost important. Therefore in order to achieve accuracy, extraction of redundant words with low or no semantic meaning must be filtered out. Such words are known as stopwords. Stopwords list has been developed for languages like English, Chinese, Arabic, Hindi, etc. Stopword list is also available for Sanskrit language.

Stop-word removal is an important preprocessing techniques used in Natural Language processing applications so as to improve the performance of the Information Retrieval System, Text Analytics & Processing System, Text Summarization, Question-Answering system, stemming etc. In this paper, a simple approach is used to design stop-word removal algorithm and its implementation for Sanskrit language. The algorithm and its implementation uses dictionary based approach. In dictionary based approach predefined list of stopwords is compared to the target text on which removal is required.

Keywords

Information Retrieval (IR), Natural Language Processing (NLP), Sanskrit, Stopword, Tokenization.

1. INTRODUCTION

Text preprocessing phase includes identification of words, phrase, sentences, stop words elimination, stemming etc. Preprocessing phase reduce size of text. Indexing, stopword elimination, stemming, phrase extraction, word sense disambiguation, query modification, and knowledge bases have also been used in Information Retrieval System to enhance performance[1]. Stop-words are frequently occurring words in a natural language which are considered as unimportant in certain natural language processing applications like Clustering, Text Summarization, Information Retrieval, etc. Almost all text preprocessing applications remove stopwords before processing documents and queries. This increases system performance. Stop-words are mainly categorized under conjunctions, prepositions, adverbs, articles of English language. In Sanskrit language most of them are classified grammatically under "*avyaya*" or the indeclinables.

Indeclinables or Sanskrit *avyaya's* are words which cannot be mutated or inflected in all genders, numbers and cases [14]. The removal of stop words in the Sanskrit language could be an important work, as its elimination reduces the feature space, thus helps in reducing time and space complexity.

In order to reduce influences on results, stopwords needs to eliminated from original text. Here generic stopword list of Sanskrit language is used, which is created using hybrid approach [2]. In Sanskrit written text , some words are common and adds no or too less meaning which adds to the content of text. A lot of CPU cycles and memory can be saved if it is removed in preprocessing phase of text. The source corpus can be greatly reduced by eliminating such words. Though the system is computationally expensive, but it gives better results. As the approach used here is dictionary based, its efficiency highly relies upon the predefined stopword list.

2. RELATED WORK

Stop words are most common words found in any natural language which carries very little or no significant semantic context in a sentence. It just carry syntactic importance which aid in formation of sentence. As a preprocessing operation it must be removed to ease further task and speedup core task in text processing. Ibrahim A [3] conducted a comparative study on the effect of stop words elimination on Arabic Information Retrieval where three stop lists viz, General Stop list, corpus based stop-list and combined stop list were used for comparative study. General stop-list performed better than the rest of the two. Ashish T, et al [4] eliminated stop-word in Gujarati language by preparing frequency list from Gujarati corpus by analyzing popular Gujarati newspapers. Riyad A, et al [5], used Finite State Machine (FSM) algorithm to eliminate stop-words for Arabic Language. Basim A, et al [6] have designed and implemented a new stop-word removal technique for Arabic language based on dedicated list and algorithm which compares stopwords if it fulfills desired string length criteria. Vijayarani S, et al[7] used Zipf's Law (Z method) for creation of stop-words. Rakholia and Saini [8] have presented a rule-based approach to dynamically identify stop words for Gujarati language. They have also deployed this approach with additional cosine similarity based Vector Space Model for information retrieval in Gujarati language [9]. Kaur J and Saini JR have presented the list of Punjabi stop words [10], its Part-of-Speech class based classification [11] and its Gurumukhi and Shahmukhi script versions [12]. Saini and Rakholia [13] have presented an analytic in-depth report on continent and script-wise divisions-based statistical measures for stopwords lists of various international Languages.

3. APPROACH USED TO REMOVE STOPWORDS.

A dictionary based approach is been utilized to remove stopwords from document. A generic stopword list containing 75 stopwords created using hybrid approach is used [2].

The algorithm is implemented as below given steps.

Step 1: The target document text is tokenized and individual words are stored in array.

Step 2: A single stop word is read from stopword list.

Step 3: The stop word is compared to target text in form of array using sequential search technique.

Step 4: If it matches , the word in array is removed , and the comparison is continued till length of array.

Step 5: After removal of stopword completely, another stopword is read from stopword list and again algorithm follows step 2. The algorithm runs continuously until all the stopwords are compared.

Step 6: Resultant text devoid of stopwords is displayed, also required statistics like stopword removed, no. of stopwords removed from target text, total count of words in target text, count of words in resultant text, individual stop word count found in target text is displayed.

4. RESULTS

The implemented algorithm was tested on approximately 2 MB of data constituting nearly 87000 Sanskrit words from different collected Sanskrit text from web and other digital media. Available 6 different documented text was feed into algorithm and the results obtained from system is listed in Table-1. Nearly 11,200 stopwords were removed thus reducing no. of words by approximately 13 %. Thus reducing feature space to an extent helps in reducing CPU cycles for data processing.

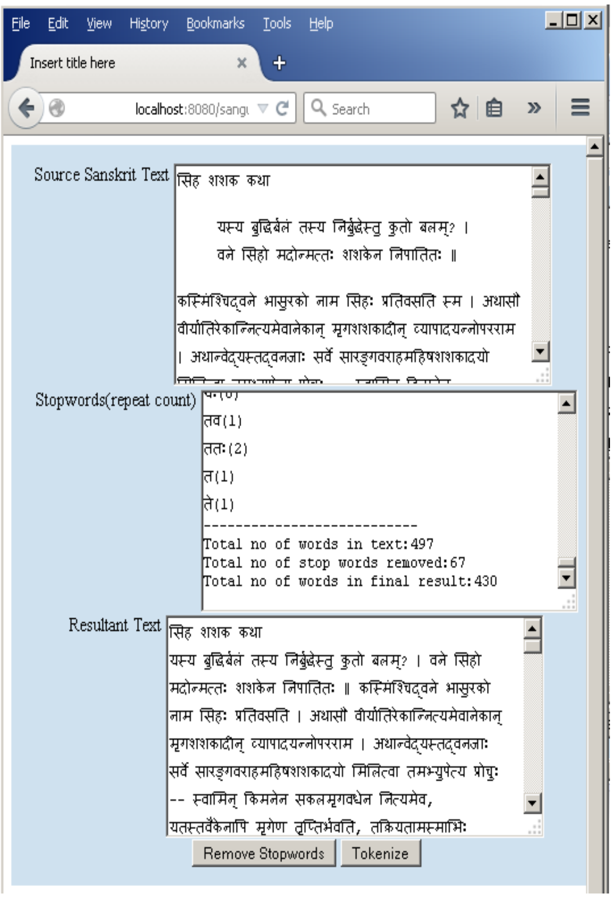
Table-1 Stopword elimination details from documented text

Sr. no.	Size in KB	Total Words in document	No. of Stopwords eliminated	Percentage removal of stopword
1	207	9031	854	9.46
2	493	24043	3574	14.87
3	491	24701	3659	14.81
4	158	5958	639	10.73
5	108	3795	440	11.59
6	556	20054	2033	10.14

The approach being dictionary based all stopwords available in dictionary were removed from Sanskrit documents. A part of Sanskrit document was analyzed manually wherein stopwords fused with other words were not identified as the current algorithm does not consider segmenting Sanskrit words. Thus efficiency attained by the algorithm is approximately 98%.

5. IMPLEMENTATION

The system has been implemented as dynamic web application. Servlet and Java Server Pages JSP API is used to develop web application. Apache Tomcat web server is utilized to host the system. For data store, MySQL database is used. The Sanskrit text is inputted in web application in UTF-8 format, the same format is also used to store data in MySQL database.



सिह शशक कथा

यस्य बुद्धिर्बलं तस्य निर्बुद्धेस्तु कुतो बलम्? ।
वने सिहो मदोन्मत्तः शशकेन निपातितः ॥

कस्मिंश्चिद्वने भासुरको नाम सिहः प्रतिवसति स्म । अथासौ वीर्यातिरेकान्नित्यमेवानेकान् मृगशशकादीन् व्यापादयन्नोपरराम । अथान्वेद्यस्तद्वनजाः सर्वे सारङ्गवराहमहिषशशकादयो

```
तव(1)
ततः(2)
त(1)
ते(1)
-----------------------------
Total no of words in text:497
Total no of stop words removed:67
Total no of words in final result:430
```

सिह शशक कथा
यस्य बुद्धिर्बलं तस्य निर्बुद्धेस्तु कुतो बलम्? । वने सिहो मदोन्मत्तः शशकेन निपातितः ॥ कस्मिंश्चिद्वने भासुरको नाम सिहः प्रतिवसति । अथासौ वीर्यातिरेकान्नित्यमेवानेकान् मृगशशकादीन् व्यापादयन्नोपरराम । अथान्वेद्यस्तद्वनजाः सर्वे सारङ्गवराहमहिषशशकादयो मिलित्वा तमभ्युपेत्य प्रोचुः
-- स्वामिन् किमनेन सकलमृगवधेन नित्यमेव,
यतस्तवैकेनापि मृगेण तृप्तिर्भवति, तक्रियतामस्माभिः

Fig-1 Screen shot of Implementation GUI

6. CONCLUSION

An issue with Sanskrit language is scarcity of digitized availability of written text. Still digitized texts from different authenticated sources were collected and used to feed into the algorithm implemented. Text from various domains like spiritual text, current information text, old and new stories, essays were considered, which were downloaded from available digitized Sanskrit text through web resources.

The Sanskrit stop-word removal technique has been tested using various Sanskrit corpora and gives better results. In information retrieval systems, effective indexing can be achieved by removal of stopwords. Finally the accuracy of algorithm for eliminating most stopwords from document depends solely on stopword list used. It can be improved drastically if implemented algorithm also follows segmentation of Sanskrit words.

7. REFERENCES

[1] Siddiqui T. and Tiwary U.S., “Natural Language Processing and Information Retrieval”, Oxford University press, 2008.

[2] Raulji J. K. and Saini J. R., “A Generic Stopword list for Sanskrit Language”, Submitted for publication.

[3] Ibrahim A, “Effects of Stop Words Elimination for Arabic Information Retrieval : A comparative study”, International Journal of Computing and Information Sciences, Vol. 4 No. 3, Dec 2006.

[4] Ashish T, Kothari M and Pinkesh P, “Pre-Processing Phase of Text Summarization Based on Gujarati Language”, International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Vol-2, Iss-4, July 2014.

[5] Riyad A, Ghassan K, Jihad J, Ahmad H and Eyad H, “Stop-Word Removal Algorithm for Arabic Language”, Information and Communication Technologies: From theory to Applications, 2004 proceedings, IEEE 2004.

[6] Basim A and Mohammad A, “Hybrid Stop-Word Removal Technique for Arabic Language”, Egyptian Computer Science Journal, Vol-30 No-1, Jan 2008.

[7] Vijayarani S, Ilamathi J and Nithya, “Preprocessing Techniques for Text Mining - An Overview”, International Journal of Computer Science & Communication Networks, Vol 5(1),7-16.

[8] Rakholia R. M. and Saini J. R., “A Rule-based Approach to Identify Stop Words for Gujarati Language”, accepted for publication in Advances in Intelligent and Soft Computing (AISC) Series, ISSN: 1615-3871, 2194-5357, 1860-0794 by Springer-Verlag, Germany.

[9] Rakholia R. M. and Saini J. R. “Information Retrieval for Gujarati Language using Cosine Similarity based Vector Space Model” , accepted for publication in Advances in Intelligent and Soft Computing (AISC) Series, ISSN: 1615-3871, 2194-5357, 1860-0794 by Springer-Verlag, Germany.

[10] Kaur J. and Saini J. R., “A Natural Language Processing Approach for Identification of Stop Words in Punjabi Language”, published in International Journal of Data Mining and Emerging Technologies; ISSN: 2249-3212 (eISSN: 2249-3220); Indian Journals, New Delhi, India; vol. 5, issue 2, November 2015; pages 114-120; DOI : 10.5958/2249-3220.2015.00015.4

[11] Kaur J. and Saini J. R., “POS Word Class based Categorization of Gurmukhi Language Stemmed Stop Words”, published in the proceedings of 1st International Conference on Information and Communication Technology for Intelligent Systems (ICTIS-2015), ISSN: 2190-3018, eISSN: 2190-3026; Springer International Publishing, Switzerland; Smart Innovation, Systems and Technologies (SIST) Series (8767), vol. 51, edition 1, pages 3-10; DOI: 10.1007/978-3-319-30927-9_1; Available Online: http://link.springer.com/chapter/10.1007/978-3-319-30927-9_1

[12] Kaur J. and Saini J. R., “Punjabi Stop Words: A Gurmukhi, Shahmukhi and Roman Scripted Chronicle”, accepted and to be published in the proceedings of National Symposium: ACM Women in Research 2016, ACM-WIR-2016, Indore, published by ACM’s International Conference Proceedings Series (ICPS), ISBN: 978-1-4503-4278-0.

[13] Saini J. R. and Rakholia R. M., “On Continent and Script-wise Divisions-based Statistical Measures for Stop-words Lists of International Languages”, accepted and to be published in the proceedings of ICIP-2016: The Society of Information Processing’s Twelfth International Multi Conference on Information Processing’s International Conference on Data Mining and Warehousing (ICDMW-2016), Bangalore; published by Procedia Computer Science, the International Journal, ISSN: 1877-0509, Elsevier, Netherlands.

[14] N. Murali, R. J. Ramasree and K.V.R.K. Acharyulu, “Avyaya Analyzer : Analysis of Indeclinables using Finite State Transducers”, International Journal of Computer Applications (0975-8887) Vol – 38, No-6, January 2012.

[15] “Sanskrit Bhagvad Gita”, Available on http://sanskritdocuments.org

[16] “Panchtantra Stories”, Available on http://sanskrit.samskrutam.com/en.literature-stories.ashx

[17] “Brahmakand, Vakyakand, Padakand”, Available on http://sanskrit.jnu.ac.in

[18] “Sanskrit Essays” Available on http://sanskrit-essays.blogspot.in