2017 IEEE 7th International Advance Computing Conference

Generating Stopword List for Sanskrit Language

Jaideepsinh K. Raulji
[1]Lecturer, Ahmedabad University, Ahmedabad, Gujarat, India.
[2]Research Scholar, Dr. Babasaheb Ambedkar Open University, Ahmedabad, Gujarat, India
Email: jaideepraulji@gmail.com

Dr. Jatinderkumar R. Saini
[1]Professor & I/C Director, Narmada College of Computer Application, Bharuch, Gujarat, India
[2]Research Supervisor, Dr. Babasaheb Ambedkar Open University, Ahmedabad, Gujarat, India
Email: saini_expert@yahoo.com

Abstract - **In the era of information burst, optimization of processes for Information Retrieval, Text Summarization, Text and Data Analytic systems becomes utmost important. Therefore in order to achieve accuracy, 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 but standard stopword list is still missing for Sanskrit language. Identifying stop words manually from Sanskrit text is a herculean task hence this paper reflects an automated stop word generator algorithm based on frequency of word and its implementation to ease the task. To fine-tune the generated list still manual intervention by language expert is required thus following a hybrid approach. The paper presents the first of its kind, a list of seventy-five generic stopwords of Sanskrit language extracted from a data amounting to nearly seventy-six thousand words.**

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

I. INTRODUCTION

Preprocessing of textual information leads to prepare data for core text mining operations. It filters out noise data from text. Stop words removal is one such method of preprocessing where frequently appearing words conveying little or no meaning are eliminated. Stopwords are words which frequently appear in text do not possess any important semantic relations. For e.g. in English language words like ‘the, in, that, those, for, of, and’ are considered as stopwords as they does not account for any key role apart from grammatical formations. Stopwords are also known as function words. Stop word removal techniques are required in many NLP activities like Information Retrieval systems wherein the words are indexed which on removal of stopwords decreases indexing space. Removal of stopwords from corpus also leads to its decreased size which increases efficiency of any NLP activities. The Sanskrit stopword list generated from this implementation will serve different NLP systems developed in future. Like other natural languages, Sanskrit due to its rich grammatical features and being mother of most Indian languages, it enjoys distinguished place in research domain like machine translation [1].

In Sanskrit sentence mostly words are delimited by space but it can also be written without space. It is possible to form whole phrase and even sentence without delimiting by space in Sanskrit. The current algorithm and implementation focuses only on the word forms which are space delimited written Sanskrit text and not on segmenting the words from phrase or sentence.

II. RELATED WORK

Semantically stopwords are considered as weak elements as it add little meaning to a sentence. They act as fillers in the sentence. Methods required to generate stopwords requires huge corpus. Its accuracy depends upon the algorithm involved, size of corpus and its subjectivity used for its generation. Alajmi A and et al [2] generated Arabic language stopword list. The list generation involved various important factors like word frequency calculation, mean and variance calculation, Entropy calculation, and Borda’s ranking. Feng Zou, et al [3] constructed Chinese stopword list using word frequency characteristic by statistical model and information model. They compared final generated list with Standard English stopwords and found most corresponding words. Ashish T, et al [4] while in creation of Text Summarization algorithm based on Gujarati language also identified and removed stopwords. Gujarati language stopwords were identified by creating a frequency list from Gujarati corpus. Sharvari G, et al [5] in their process of extraction of rootwords for Devnagri script also identified stopwords especially for Marathi language. Hassan S, et al [6] generated English language stopword list using contextual semantics methodology for sentiment analysis of Twitter data. Deng Na, et al [7] generated Chinese language stopword list which would help them for documents related to Chinese patents. Joshi H, et al [8] eliminated stopwords for gaining better accuracy in information retrieval process in Gujarati text documents. Hakan A, et al [9] proposed method for generating stopwords which is domain specific. The automatically generated stopword list was tested using maximum posteriori probability estimation of keyword distribution using bag of words model, implementing with Bayesian natural language classifier for webpage. The generated stopword list by their model was compared to available standard generic stopword list for English language. Sinka M, et al, [10] created

978-1-5090-1560-3/17 $31.00 © 2017 IEEE
DOI 10.1109/IACC.2017.155

stopword list using word entropy methodology using random webpages and bank search dataset. Asubiaro, Toluwase V, [11] employed entropy based algorithm to identify stopwords for Yoruba language text. A word whose entropy was greater than 0.6 but not a noun was considered as stopword. Walaa M, et al [12] generated stopword list from Online Social Network (OSN) corpora like Twitter, Facebook etc for Egyptian Dialect (ED). Kaur J and Saini JR have presented the list of Punjabi stop words [21], its Part-of-Speech class based classification [22] and its Gurumukhi and Shahmukhi script versions [23]. Saini and Rakholia [24] have presented an analytic in-depth report on continent and script-wise divisions-based statistical measures for stopwords lists of various international Languages. Rachel T WL, et al [13] proposed a method which automatically generated stopword list using term-based random sampling approach developed by them. The novel approach is optimal and has lower computational overhead than the former ones.

III. Approach Used To Extract Stopwords.

Different algorithms, parameters and metrics can be considered to extract stopwords. But most concepts revolve around frequency of existence of such words in a source text. In this paper stopwords are extracted based on frequency of such words found in inputted text using automated algorithm. This generated list contained nouns along with potential stopwords which were manually removed from the list, also few potent stopwords were added manually thus following a hybrid approach. An issue with Sanskrit language is scarcity of digitized availability of 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. Approximately 2 MB of data containing total of 75928 words is used to feed the system. Following lines and flowchart Fig-1 depicts the implemented algorithm.

The Algorithm

1. Tokenizing stream of data to words delimited by space and new line character.
2. Initializing pivot and index word for comparison. All the words in a stream are compared to one base word. Such base word is known as pivot and others are known as index words.
3. If match is found, stopword count is incremented and eventually pivot word is compared to rest of the index tokens.
4. At the end of complete comparison, frequency is calculated based on no. of occurrences verses Total no. of words in data stream.
5. If frequency percentage crosses user defined fixed threshold value, the word is considered as stopword and inserted/updated in database.
6. Again Step 2 is repeated till Pivot reaches the end of word stream.

IV. Results

The implementation of algorithm resulted into extraction of stopwords including some nouns. Those nouns were removed and few potent stopwords were added manually which resulted to 75 stopwords listed in Table-1.

Formula

Percentage Availability of Stopword (**PSW**)

PSW= (No. of stopword in stream / Total no. of
words in stream) x 100

The threshold value (PSW) was set to 0.25% to consider word as member of Sanskrit stoplist which was considered after feeding various Sanskrit texts of different domains to the algorithm. Higher value above threshold leads to increase in noun words and lower eliminates potential stopwords. Thus after setting threshold value to 0.25%, 190 stop words were extracted including domain specific nouns within text. These nouns were manually removed leaving 64 stopwords. Finally list with 75 stopwords was created by adding 11 stopwords manually.

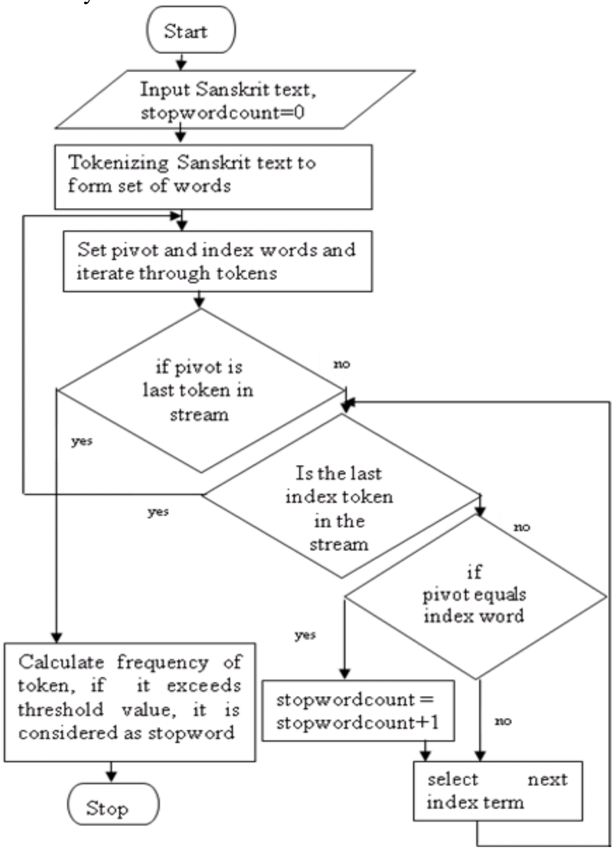
Start
Input Sanskrit text, stopwordcount=0
Tokenizing Sanskrit text to form set of words
Set pivot and index words and iterate through tokens
if pivot is last token in stream
no
yes
Is the last index token in the stream
yes
no
if pivot equals index word
yes
no
Calculate frequency of token, if it exceeds threshold value, it is considered as stopword
stopwordcount = stopwordcount+1
select next index term
Stop

Fig. 1. Flow graph of the Algorithm

TABLE-1

Sr. No.	Sanskrit Stopword	Transliterated word	Meaning
1	अतः	atah	therefore, so, hence
2	अत्र	atra	here
3	अथ	atha	but, else
4	अपि	api	but, also, even, too , and
5	अयं	ayam	this
6	अयम्	ayam	this
7	अस्ति	asti	existent, present
8	अस्मि	asmi	am
9	अस्य	asya	is, in
10	अहं	aham	I
11	अहम्	aham	I
12	आम्	aam	yes
13	इति	iti	to, thus
14	इदं	idam	this, here, now
15	इदम्	idam	this, here, now
16	इमे	ime	these
17	इयं	iyam	this, that
18	इयम्	iyam	this, that
19	एतत्	etat	this
20	एतद्	etad	this, here, now, thus, so
21	एते	ete	these
22	एव	eva	only
23	एवं	evam	thus
24	एष	es	this
25	एषा	esaa	this
26	कं	kam	yes, well, bliss
27	कः	kaha	who
28	कथं	katham	how
29	का	kaa	who
30	कानि	kaani	what
31	किं	kim	what
32	किम्	kim	what
33	कुत्र	kutra	where
34	के	ke	who
35	क्वचित्	kvachit	somewhere, somewhat
36	खलु	khalu	now
37	च	cha	and, also
38	तं	tam	him, they
39	ततः	tatah	therefore, later, then
40	तत्	tat	that
41	तत्र	tatra	there
42	तदा	tadaa	then, always
43	तनि	tani	they all
44	तव	tava	yours
45	तस्य	tasya	his
46	तस्याः	tasyaah	her
47	तु	tu	and, but
48	ते	te	they
49	तेन	tena	therefore, thus, there
50	तौ	tau	they
51	त्वम्	tvam	you
52	न	na	no
53	नु	nu	at once, now
54	नो	no	not
55	ननु	nanu	indeed
56	परन्तु	parantu	but
57	मम	mama	my, I
58	मा	maa	no
59	मे	me	my
60	य	ya	mover, goer
61	यत्	yat	that
62	यत्र	yatra	when, wherever
63	यथा	yathaa	than, as, that
64	यदा	yada	when
65	यदि	yadi	if
66	युयं	yuyam	you all
67	येन	yena	as, since, because
68	वयं	vayam	we
69	वा	va	or
70	स	sa	he
71	सः	sah	he
72	सह	saha	together
73	सा	saa	she
74	स्म	sma	always, surely
75	हि	he	because, for

List of Generated Sanskrit Stopwords

Almost all stopwords in Sanskrit are indeclinables grammatically. Indeclinables are words which are grammatically not inflected. In Sanskrit, *avyaya* i.e.- the indeclinable, plays an important role in the construction of a sentence and can be used as preposition, interjection, particle, conjunction or an adverb[14].

To best of our knowledge Sanskrit stopword list is still not available. Hence, the list presented here is released for public use for future NLP tasks in Sanskrit language.

V. Conclusion

Though Sanskrit is considered as important language in Indo-European language family, still lot of work is required to explore the potential of this language to open vistas in computational linguistics domain. Identification of stopwords in Sanskrit language may help researchers for various text preprocessing activities such as Information Retrieval, Text Summarization, spelling normalization, stemming, lemmatization, phrase matching, study of prosody in Sanskrit written text.

The generated list can still be improved if word is split, also known as sandhi *vichheda* (split) because Sanskrit sentence can also syntactically written by fusing words.

.

References

[1] Raulji J. K. and Saini J. R., “Sanskrit Machine Translation Systems : A Comparative Analysis”, International Journal of Computer Applications, Vol 136, Issue-1, Feb 2016.

[2] Alajmi A., Saad E. M. and Darwish R. R., “Toward an ARABIC Stop-Words List Generation”, International Journal of Computer Applications, Volume 46-No. 8 May 2012.

[3] Feng Z, Fu Lee W, Xiaotie D, Song H and Lu Sheng W, “Automatic Construction of Chinese Stop Word List”, Proceedings of the 5th WSEAS, International Conferences on Applied Computer Science, Hangzhou, China, April 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) Volume-2, Issue-4, July 2014.

[5] Sharvari G, Bakal J W and Sagar K, “Extraction of Root Words using Morphological Analyzer for Devanagari Script”, International Journal Information Technology and Computer Science, Jan 2016.

[6] Hassan S, Miriam F and Harith A, “Automatic Stopword Generation using Contextual Semantics for Sentiment Analysis of Twitter”, Knowledge Media Institute, The Open University, United Kingdom.

[7] Deng N and Chen X, “Automatically Generation and Evaluation of StopWords List for Chinese Patents”, TELKOMNIKA Vol 13, No 4, Dec 2015.

[8] Joshi H, Pareek J, Patel R and Chauhan K “ To stop or not to stop – Experiments on stopword elimination for information retrieval of Gujarati text documents”, IEEE, Engineering (NuiCONE), Nirma University, 2012.

[9] Hakan A and Sirma Y, “An automated domain specific stopword generation method for natural language text classification”, Innovations in Intelligent Systems and Applications (INISTA) , 2011, International Symposium publisher IEEE.

[10] Sinka M and Corne D, “Towards Modernised and Web-Specific Stoplist for Web Document Analysis”, International Conference on Web Intelligencce 2003, published by IEEE computer society, Washignton DC USA 2003.

[11] Asubiaro, Toluwase V, “Entropy-Based Generic Stopwords List for Yoruba Texts”, International Journal of Computer and Information Technology, Volume 2, Issue - 5, Sept 2013.

[12] Walaa M, Ahmed Y and Hoda K, “Egyptian Dialect Stopword List Generation from Social Network Data”,Egyptian Journal of Language Engineering, Vol 2, No. 1, April 2015.

[13] Rachel T, Ben H and Ladh O, “Automatically building a Stopword list for an Information Retrieval System”, 5th Dutch-Belgium Information Retrieval Workshop (DIR), 2005, Utrecht, The Netherlands.

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

[15] Anand R and Harikrishna M, “Transliteration based Search Engine for Multilingual Information Access”,Proceedings of CLIAWS3, Third International Cross Lingual Information workshop, Colarado, June 2009.

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

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

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

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

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

[21] 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

[22] 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

[23] 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.

[24] 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