National Conference on Engineering Innovations and Solutions (NCEIS – 2018)
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 6 | ISSN : 2456-3307

Effect of Dynamic Stoplist on Keyword Prediction in RAKE

Avinash Bhat, Chirag Satish, Nihal D'Souza, Nikhil Kashyap
Department of Computer Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India

ABSTRACT

Keywords which we define as a sequence of words that provide a condensed representation of the document in question. These keywords are vital in numerous applications from web search engines to abstractive text summarization. Rapid Automatic Keyword Extraction (RAKE) [1] is an unsupervised, domain and language independent method for extracting keywords from documents. RAKE is based on the simple observation that keywords seldom contain stop words – such as and, of and the. RAKE uses a list of stop words to split the document text into candidate keywords. The list of stop words or stoplist is static. In this paper, we make the stoplist dynamic, in that, stop words, that do not currently belong to the stoplist but are identified as potential stop words for the given document are added to the stoplist. Consequently, every document has a unique stoplist. We compare the performance of our implementation to the standard RAKE implementation on Wikipedia articles.

Keywords:RAKE, Keyword extraction, Stopwords, Dynamic, Wikipedia

I. INTRODUCTION

With respect to text documents, keywords refer to phrases which paint a holistic picture of the article to the reader. The increase in the number of documents on the web without a list of keywords has necessitated the need for tools that automatically generate keywords for the given input document. Keyword extraction is also an important task in problems like Natural Language processing, text mining and summarization. A typical keyword extraction tool has three main modules:

1. Selection of candidate keywords: Using stop words, phrases which can potentially be the keywords of the document are identified.
2. Property evaluation: Every candidate keyword is evaluated based on a number of factors such as adjacency, frequency, location in the document.
3. Selecting keywords: All candidates can be scored either by uniting the properties into a formula or by using machine learning techniques to calculate the probability of a candidate being a keyword.

Our study is restricted to Rapid Automatic Keyword Extraction (RAKE) – an unsupervised, domain and language independent keyword extraction tool proposed in [1]. RAKE uses a static stop list to break the document down into a list of stopwords. A method to automatically generate the stoplist from a set of documents where the keywords are defined, called the Keyword Adjacency (KA) stoplist has also been proposed. It is based on the insight that words adjacent to, and not within the keywords are likely candidates for stop words. The frequency of each word appearing adjacent to the keyword is tabulated, and words which occurred more frequently within keywords than adjacent were excluded. This method was compared with the stop list generated by Term Frequency (TF). It was concluded that the KA stoplist outperformed the TF stoplist, and moreover the best TF stoplists underperforms compared to the worst KA stoplist. However, this method requires a document with pre-defined stop words, and the stoplist only makes considerable difference in keyword prediction if the KA algorithm is run on several documents.

Our work focuses on studying the effect of keyword prediction in RAKE, when every document has a unique stoplist to reflect its characteristics, and the improvements in prediction that arise compared to the standard RAKE implementation - that uses the NLTK stoplist.

II. RELATED WORK

Stop words in review summarization using TextRank by Sonya RapintaManalu, 2017 presents a comparison of automatic review summarization with and without stop words. An extractive, unsupervised graph-based ranking model TextRank is employed to highlight the differences between both approaches. Experimental results on 50 sample reviews have shown that the usage of stop words removal can be impactful in determining the result of review summarization, which suggests that depending on the user requirements, it should be considered whether stop words removal needs to be performed or not. [2]

Stop-words in keyphrase extraction problem by S. Popova, L. Kovriguina, D. Mouromtsev, I. Khodyrev, 2013. Keyword extraction problem is one of the most significant tasks in information retrieval. High-quality keyword extraction sufficiently influences the progress in the following subtasks of information retrieval: classification and clustering, data mining, knowledge extraction and representation, etc. The research environment has specified a layout for keyphrase extraction. However, some of the possible decisions remain uninvolved in the paradigm. In the paper the authors observe the scope of interdisciplinary methods applicable to automatic stop list feeding. The chosen method belongs to the class of experiential models. The research procedure based on this method allows to improve the quality of keyphrase extraction on the stage of candidate keyphrase building. Several ways to automatic feeding of the stop lists are proposed in the paper as well. One of them is based on provisions of lexical statistics and the results of its application to the discussed task point out the non-gaussian nature of text corpora. The second way based on usage of the Inspec train collection to the feeding of stop lists improves the quality considerably. [3]

Wilbur, W.J. and Sirotkin, K., 1992. The automatic identification of stop words. Journal of information science, 18(1), pp.45-55 [4] defines a stop word as "a word which has same likelihood of occurring in those documents not relevant to the query as in those documents relevant to the query". This paper follows the TF-IDF model, first by calculating the similarity between two documents, and then calculating the number of words which occur in both the documents. This effort is done in order to explore the effect of stop words in information retrieval.

Silva, C. and Ribeiro, B., 2003, July. The importance of stop word removal on recall values in text categorization. In Neural Networks, 2003. Proceedings of the International Joint Conference on (Vol. 3, pp. 1661-1666). IEEE [5] - A comparison on accuracy and precision-recall values corresponding to a support vector machine states that Stop word removal removes information that could mislead the learning machine. The test conditions which were followed in the paper were based on frequency of the words, existing stop words and using stemming.

Yao, Z. and Ze-wen, C., 2011, March. Research on the construction and filter method of stop-word list in text preprocessing. In Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on (Vol. 1, pp. 217-221).

IEEE defines certain rules for construction of stop word list and also compares the efficiency of different filters which are used to detect and eliminate the stop words from a given corpus. [6]

"Automatically building a Stopword list for an information retrieval system" by Rachel Tsz-Wai Lo et. al. [7] evaluates different methods for generating the stop list for a given collection of documents automatically. An innovative approach called the term back random sampling is introduced - which determines how informative a term is, to aid with the stop list generation. It is also shown that the best results can be obtained by combining the classical stop word list with the stop words generated by term back sampling method.

"On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter" by Hassan Saif, Miriam Fernandez, Yulan He, Harith Alani [8] concludes that pre-compiled stoplist negatively impacts sentiment classification whereas dynamic generation of stop list result in better performance.

III. PROPOSED SYSTEM

The RAKE algorithm, makes use of a static stop word list which is common to a variety of documents [1]. The proposed system suggests that generation of stop words which are specific to document can improve the efficiency of the algorithm. One of the methods which can be followed to extract stop words from the document is based on their lexical categories. Intuitively, stop words can be tagged as conjunctions, determiners and so on. This information is extracted using a part of speech tagger to generate the stop word list which is exclusive for the given document.

Algorithm

```
lexCategory = [adverb, conjunction, determiner, article, pronoun,…]
for everyWord in document
        if wordCategory in lexCategory
                append word to stopWordList
```

This is done with the help of NLTK's part of speech tagging module [9]. This returns the tag for every word in the document corpus. A list of categories of all possible stop words is constructed and then compared with all the words in document, which gives the set of stop words.

Further, this list of stop words, is sent as an input to the RAKE algorithms to get the key phrases and score.

IV. RESULT AND ANALYSIS

In this section, we highlight the working of the algorithm taking a few examples. For simplicity, we consider two general categories – Politics and Sports. Under each of these two categories, we again consider two personalities each. Under the category of politics, we consider – Narendra Modi and Justin Trudeau. For sports – Roger Federer and Lewis Hamilton.

We have considered two test cases of stop words that will be compared. The first test case is the standard set of NLTK stop words that Rake has included in its package. These stop words are fed into the Rake algorithm and a set of words with their relevance score is obtained. Similarly, for test 2, again a set of words with their relevance score is obtained, but the code is modified to also accept a set of dynamically obtained stop words along with the NLTK stop words. These two cases are compared using graphical analysis and will help prove that dynamically obtained stop words help in obtaining higher relevance score words. This results in a higher accuracy of summarization for the personality, or any subject for that matter.

The line graph is drawn according to the relevance score (from Rake) vs. the top words that were common between two test cases considered for stop words – Only NLTK stop words and NLTK stop words plus dynamically obtained stop words.

Considering the personality of Justin Trudeau, we obtain the graph in Fig. The dotted line represents the graph obtained in the case when only NLTK stop words are considered and the solid line represents the graph obtained when both NLTK and dynamically obtained stop words are considered.

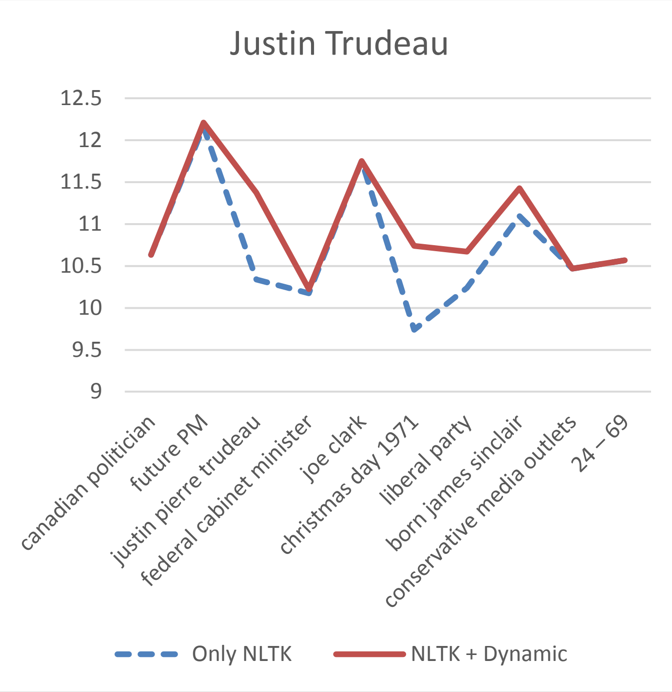

Figure 1: Line graph of top relevant words vs. score for 'Justin Trudeau'

Words such as 'christmas day 1971' and 'justinpierretrudeau' clearly show a higher relevance value in the latter case over the former. These words are significant to the subject because the word 'christmas day 1971' describes his birthday and 'justinpierretrudeau' is his full name. Words such as 'canadianpolitican' and 'federal cabinet minister' have no increase in their relevance scores.

These characteristics can also be highlighted in a similar personality figure under the same category of politics – Narendra Modi.

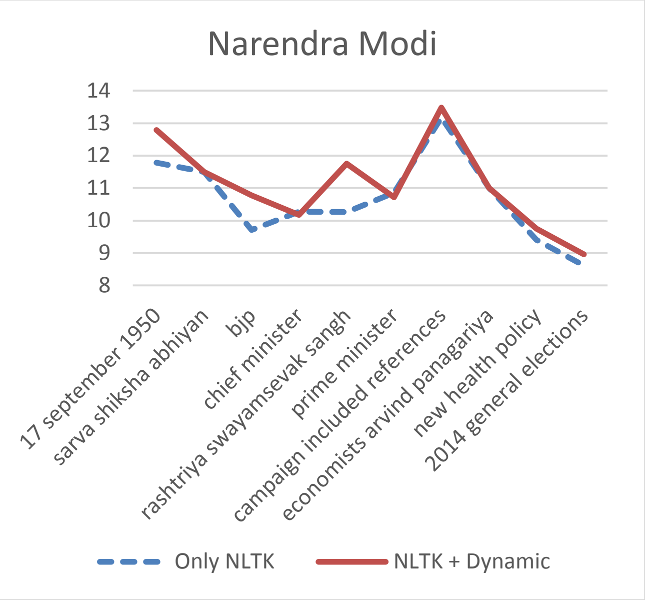

Figure 2: Line graph of top relevant words vs. score for 'Narendra Modi'

Here we notice that words such as '17 september 1950' and 'bjp' have a higher relevance score in the case of dynamic plus NLTK stop word list algorithm, as compared to only the NLTK stop words list algorithm. Other words such as 'sarvashikshaabhiyan' and 'prime minister' have negligible increase in their relevance scores.

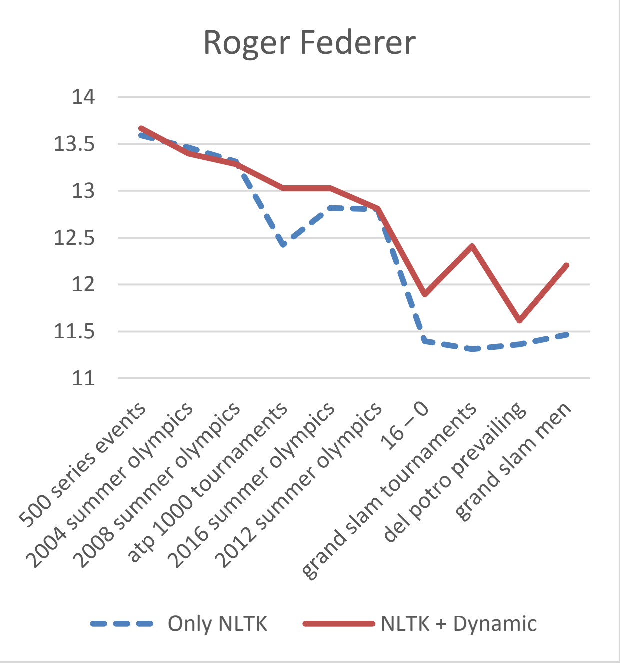

Figure 3: Line graph of top relevant words vs. score for 'Roger Federer'

Under the category of Sports, we take the example of Roger Federer. Here we notice that words such as ‘atp 1000 tournaments’ and ‘grand slam tournaments’ have a higher relevance in the case we use NLTK + Dynamic stop word list as compared to only the NLTK list.

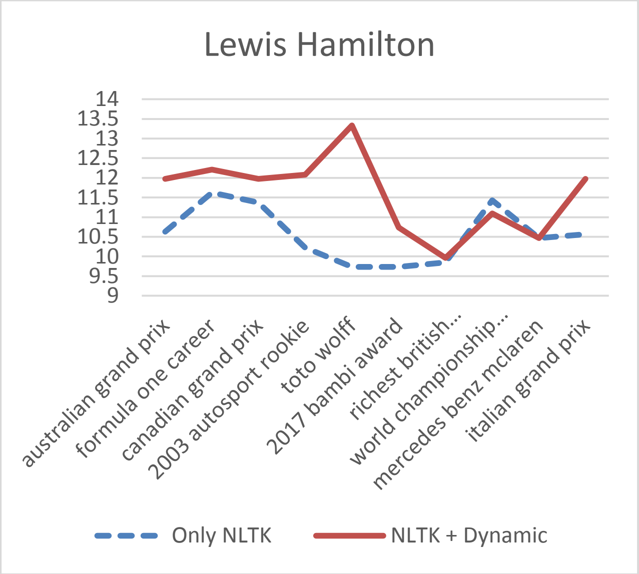

Figure 4. Line graph of top relevant words vs. score for ‘Lewis Hamilton’

Similarly, we can draw the same conclusions with the personality of ‘Lewis Hamilton’ where terms such as ‘toto wolff’ and ‘2003 autosport rookie’ take a higher relevance value when dynamic stop words are considered along with the NLTK list of stop words.

We notice that important details about the personality tend to obtain a higher relevance score in the second case as compared to the first. This happens due to the fact that the second test case offers a larger data set of stop words hence eliminating more common words particular to that category. In doing so, information pertaining particularly to the subject tends to be given a higher score amongst the list of remaining words. This translates to the words holding information about the personality that pertain to individual subject either gaining a higher score or remaining the same.

V. CONCLUSION

The variation and different styles of writing makes keyword extraction a very difficult problem. While the standard NLTK stoplist is universally applicable, it isn’t perfect. As we have shown, a dynamic stoplist that captures properties of the article along with the standard stoplist produces better results in RAKE. The downside to our system is that using words belonging to the document in the stoplist results in omission of certain phrases that would otherwise have been categorized as candidate keywords. We can add additional conditions such as the location of the keyword in the document, frequency and relevance to the document to improve the stoplist. In conclusion, a dynamic stoplist produces better results but comes with the tradeoff that certain phrases are omitted from consideration as keywords.

VI. ACKNOWLEDGMENT

We would firstly like to thank Prof. H D Phaneendra, Head of the Department of Computer Science and Engineering, The Natioanl Institute of Enginering, Mysore for giving us the support and opportunity to work on this paper. We also thank Dr.Anitha C for her guidance. We also extend our warm regards to all the faculty members of the department for their constant encouragement and invaluable advice.

VII. REFERENCES

1. Rose, S., Engel, D., Cramer, N. and Cowley, W., 2010. Automatic keyword extraction from individual documents. Text Mining: Applications and Theory, pp.1-20.https://www.researchgate.net/profile/Stuart_Rose/publication/227988510_Automatic_Keyword_Extraction_from_Individual_Documents/links/59edf51fa6fdccbbefd5434a/Automatic-Keyword-Extraction-from-Individual-Documents.pdf
2. Stop words in review summarization using TextRank by Sonya RapintaManalu, 2017 http://ieeexplore.ieee.org/document/8096371/
3. Stop-words in keyphrase extraction problem by S. Popova, L. Kovriguina, D. Mouromtsev, I. Khodyrev, 2013 http://ieeexplore.ieee.org/document/6737953/

4. Wilbur, W.J. and Sirotkin, K., 1992. The automatic identification of stop words. Journal of information science, 18(1), pp.45-55
5. - Silva, C. and Ribeiro, B., 2003, July. The importance of stop word removal on recall values in text categorization. In Neural Networks, 2003. Proceedings of the International Joint Conference on (Vol. 3, pp. 1661-1666). IEEE http://www.sciencedirect.com/science/article/pii/S0167739X10002554
6. Yao, Z. and Ze-wen, C., 2011, March. Research on the construction and filter method of stop-word list in text preprocessing. In Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on (Vol. 1, pp. 217-221). IEEE
7. "Automatically building a Stopword list for an information retrieval system" by Rachel Tsz-Wai Lo et. al. http://terrierteam.dcs.gla.ac.uk/publications/rtlo_DIRpaper.pdf
8. "On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter" by Hassan Saif, Miriam Fernandez, Yulan He, Harith Alani
9. NLTK http://www.nltk.org/_modules/nltk/tag.html