Searching and Ranking Relevant Textual Data: Relevance Filter Application

Quynh Tu Hoang

1. Introduction

Finding relevant data in unstructured massive amount of text from different sources with different methods and means has been the core of Information Retrieval (IR) research for more than 60 years (Chu 2003). While IR researchers has made substantial breakthroughs in their endeavors, there are still some problems that can cause discrepancies in the matching of search queries and the information represented in the database or corpus. One of the main problems is that most of the approaches are based on the assumption that users can foresee the exact words and phrases that will be used in the documents they will find relevant. It becomes more apparent and problematic when the database or corpus comprises of documents written by different authors and in different styles.

Retrieving documents that are relevant to a particular topic is a fundamental task in conducting research using textual data from various sources. Research in the social sciences and humanities domains has been substantially benefited from the availability of massive data on the Web, especially user-generated content, to study human behavior and social phenomena. However, user-generated content, i.e. textual data from social networking sites and forums, is in most cases unstructured and considerably varied in style, which poses a significant challenge for researchers to find relevant data in the corpus.

This paper presents an approach to retrieving information that is relevant to a topic from a large corpus of unstructured documents, which is a fundamental step prior to

conducting any quantitative or qualitative analysis. The paper aims to go beyond the presentation of architecture and account for the rationale driven the design of the application as well as the important issues that researchers should consider and reflect on when using the application. In order to situate the proposed application, I will first introduce some of the approaches to extracting relevant textual data from a large corpus, which led to the development of my approach. The rest of the paper is dedicated to introducing the Relevance Filter application and providing examples of data filtering and analysis it enables. The application is to measure to what extent a document is relevant to the topic being researched. Before concluding, I will propose the future development of the application to improve the outcome of the relevance score measurement.

2. Reasons that led to the development of the application

The main motivation for this experiment and the development of the application is to use computational techniques to find relevant texts in corpora because it is observed that classifying texts by relevance could become a huge and time-consuming task in a research project, especially when the corpora are large datasets from social media platforms. While web search engines have been extensively developed and with the use of machine learning algorithms, there are still relatively few free relevance ranking programs for researchers. Therefore, I carried out this experiment to find an efficient approach for retrieving relevant data from a corpus and develop it to a tool that is flexible and easy-to-use.

One of the earliest approaches for finding information is to index a document collection and to search for keywords (words and phrases that are relevant to the topic in question) or more complex search expressions in Boolean logic in order to find documents matching the criteria. In their influential study of retrieval effectiveness, Blair and Maron (1985, p. 295) pointed out that:

> "The belief in the predictability of the words and phrases that may be used to discuss a particular subject is a difficult prejudice to overcome… Stated succinctly, it is impossibly difficult for users to predict words, word combinations, and phrases that are used by *all* (or most) relevant documents and *only* (or primarily) by those documents."

We can assume that researchers have certain knowledge of the topic they are investigating, and therefore can make some good speculations on keywords. However, their knowledge may come from academic documents or mainstream sources in which language and style can be vastly different from those in user-generated content on the web. Consequently, the result may contain zero hit if the queries defined by researchers as relevant to their topic of research do not exist in the corpus.

Another approach is to choose a number of good representative documents that are highly relevant to the topic and then extract keywords from them. These keywords are subsequently used to search for other relevant documents in the corpus. However, this approach still potentially hinders important keywords that might not exist in the chosen representative documents. This is especially the case when the documents in the corpus are varied in style because they are written by different authors from different backgrounds.

The fundamental problem of both approaches is that the output is not ranked in the order of relevance, especially when the data corpus is large. Therefore, I propose an approach that would solve the above-mentioned problems. In the following section, I will explain the main mechanisms of the application I developed and the reasonings behind them.

3. Relevance Filter – An application for ranking relevance of textual data

3.1 Background and Overview of the application

The idea for my algorithm is simply to find documents that contain the keywords defined as relevant to the topic by the researcher. To minimize and even eliminate the discrepancies in matching the keywords and documents in the corpus, it is important that the keywords are extracted from the corpus itself. While that is the main point of my approach, it is necessary to understand the whole procedure of the application as finding textual data that satisfies the information need of a research project is not simple but consist of multiple layers.

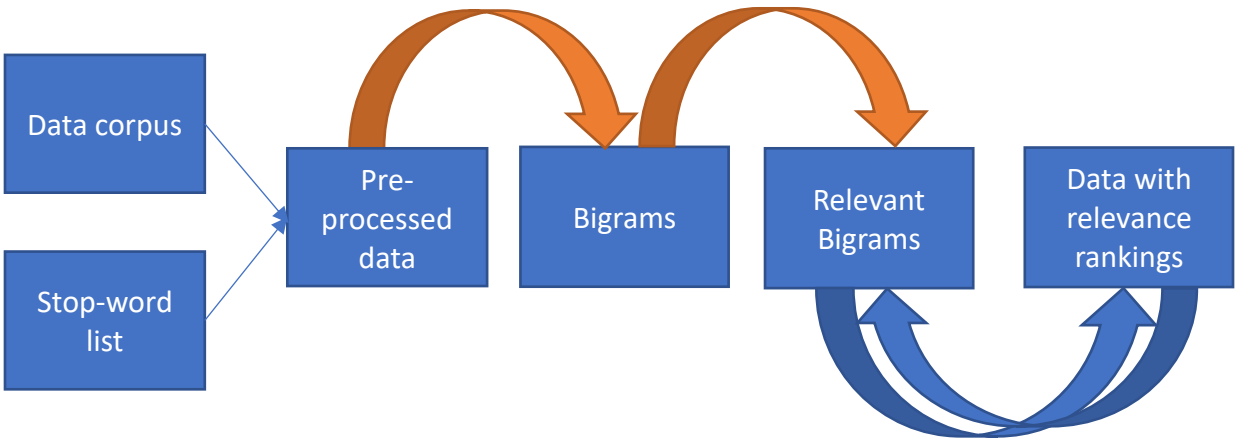

Figure 1. The procedure of the Relevance Filter application

At the first step, the data corpus is uploaded to the database of the application in the form of a UTF-8 comma-separated values (CSV) file. After choosing the column that contain the documents to be ranked by relevance, the documents go through the data pre-processing phase to reduce noise and prepare data for the next phase, which is generating a bag-of-bigrams from the documents. Subsequently, the researcher chooses the bigrams that they think are likely to be relevant to the topic of their research as queries for searching the documents containing these bigrams.

The initial experiment was retrieving only documents including the defined-as-relevant bigrams. Using this method, I narrowed down from the initial 1,991 documents to 928 documents that include at least one of the bigrams. While such

reduction of documents saved us time and effort in reading and selecting relevant documents, it was still inefficient as there was no indication of to what extent a document is relevant to the topic. This approach requires researchers to give equal time of reading to all the documents that include the selected bigrams, which was still extremely time-consuming. Therefore, I added a scoring algorithm to the program to determine how relevant a document is to the topic. The outcome for users is a downloadable UTF-8 CSV file which is the original file they uploaded to the database with an added column of correspondent relevance scores. This approach, therefore, retains the whole dataset which may include other statistic columns important for further analysis.

The textual data processing of the application uses Natural Language Toolkit (NLTK), which is a library of open-source and independent modules for building natural language processing (NLP) programs in Python. The application is aimed to deal with different languages, and different character sets. Therefore, it is essential to use appropriate encoding for processing texts that use non-ASCII character sets. I choose to enforce the uploading of documents with UTF-8 encoding because it is the most common encoding in the Unicode family, which aims to support all written languages on the web. According to a survey by W3Techs (2019), as of February 2019, UTF-8 is used by 93.1% of all the websites.

3.2 Preprocessing data

Researchers must impose some preliminary restrictions to reduce the dimensionality of the data to a manageable level. A common first step is to strip out elements of the raw text other than words. In order to get a cleaned and normalized text before tokenizing the document, the application removes all punctuations and special characters from the corpus. However, I made a conscious choice to not convert all letters to lower case because in some languages, such as Vietnamese, it is very common that personal names are just regular words signified by capitalization. In this case, converting all letters to case would dismiss the distinguish between

personal names and regular words. As my experiment used Vietnamese datasets, the regular expression used to normalize and transform the datasets is specifically formulated to deal with Vietnamese characters:

```
regex = re.compile("[^a-z0-9A-
Z_àáãạảăắằẳẵặâấầẩẫậèéẹẻẽêềếểễệđìíĩỉịòóõọỏôốồổỗộơớờởỡợùúũụủưứừửữựỳỵỷỹýÀÁÃẠẢĂẮẰẲẴẶ
ẠẢẤẦẨẪẬÈÉẸẺẼÊỀẾỂỄỆĐÌÍĨỈỊÒÓÕỌỎÔỐỒỔỖỘƠỚỜỞỠỢÙÚŨỤỦƯỨỪỬỮỰỲỴỶỸÝ ]")
```

Figure 2. The regular expression rule of the application

3.3 Stopwords

Cleaning textual data also involves removing a subset of words that are either very common or very rare. Very common words, often called “stop words,” include articles (“the”, “a”), conjunctions (“and”, “or”), forms of the verb “to be,” and so on. Stopwords, by definition, typically have low discrimination power or convey relatively little meaning on their own (Lo et al. 2005). Hans Peter Luhan (1957) introduced the first approach to divide words in natural language texts into keyword terms and stopwords. A common practice is to exclude stopwords based on a pre-compiled stop-word list. The pre-compiled stop-word lists are argued to be too small, too generic and outdated because they do not cover emerging stopwords on the web (Lo et al. 2005, Sinka and Corne 2003, Yang 1995). Therefore, several methods have been developed to automatically generating stop-words that have recently emerged (Trumbach and Payne 2007, Makrehchi and Kamel 2008, Lo et al. 2005).

Since the goal of my application is to enable filtering textual data in any language that is supported by UTF-8 encoding, I leave it up to users to use their own stop-word list. The pre-compiled stop-word list of a specific language can be found by using a web search engine. However, it is highly advised that researchers understand what should be included in the stop-word list. For example, for data retrieved from the web, researchers might want to add some emerging stopwords, such as “www”, “http”, “https”, “com”, “tinyurl”, “bitly”, etc. In addition, researchers might want to add context-specific stop words. For example, if the corpus consists of many

advertisements which are irrelevant to the topic, the researcher should analyze the patterns of the advertisements, find words that commonly appear in the advertisements, and add them to the stop-word list.

The stop-word list must be uploaded to the database of the application also in the form of a UTF-8 CSV file. The stop words in the list will be stripped out from the corpus before the application generates the bigrams list to reduce noise in the list, and ensure that the program returns mostly important bigrams.

3.4 A bag-of-bigrams

n-gram is a contiguous sequence of n items from a given sample of text or speech. Using Latin numerical prefixes, an n-gram of size 2 is a "bigram". In other words, a bag-of-bigrams is a collection of two successive words. Fürnkranz's experiment (1998) which used an algorithm based on the APRIORI algorithm for efficiently generating features using term frequency and document frequency as criteria came to a conclusion that word sequences of length 2 or 3 were most useful, while longer sequences reduced categorization performance. Therefore, the approach I propose is to generate a bag-of-bigrams from the corpus, which is subsequently scrutinized by researchers to select the bigrams that are relevant to the topic (keywords).

There can be no instruction on a good number of bigrams to be selected as relevant; however, it is advisable that the researcher work carefully on the list of relevant bigrams, and even go back and forth with it until the result seems comprehensive. When selecting bigrams as keywords, researchers should be aware of the correlation between the frequency of the bigrams and the resulting relevance scores in relation to the whole dataset. If the keywords are too generic, it is unlikely that the relevance ranking would be useful as too many documents would hit a match with the keywords. The application allows researchers to go back to edit their selections of bigrams if the outcome is unsatisfactory provided that researchers keep the window browser open and make their edits within three hours.

To optimize user experience and loading time, I limit the list of bigrams to 1,000 bigrams with the highest frequency. This would require that the corpus has more than just a few relevant data. If the data has more irrelevant data than relevant data, the application would not be useful as it is unlikely that many relevant bigrams would be in the 1,000 most-repeated bigrams. An important note for research focusing on linguistic aspect is that the bigrams generated from the corpus are potentially useful in analyzing linguistic style of a specific individual/group or comparing linguistic features between different individuals /groups.

3.5 Relevance Score

The final step of the program is generating relevance score based on how many times a match is found between the texts in the corpus and the keywords selected by the researcher. When a match is found, the document is given one additional score to its "relevance ranking". Matching is the fundamental mechanism in retrieving relevant information. The ultimate goal for quality relevance ranking is to minimize the discrepancies can that occur during the whole process. Using selected bigrams generated from the corpus as the source of keywords for the matching is the main method to minimize the discrepancies. However, the core of the approach I propose is still the researcher's knowledge of the topic being diagnosed and the data corpus itself. It is important that the researcher read through the documents in the corpus to get a sense of the language and style used in the documents, which bigrams would be relevant to the topic they are investigating, and which bigrams would be too generic. Besides giving indication of to what extent a document might be relevant to the topic, the ranking score of a document give researchers a hint of how much important the document might be.

3.6 Security measures

The application's traffic is encrypted from SSL, while certificates are provided and managed through the Let's Encrypt service[1]. The application itself consists of various virtual servers over a webserver proxy, meaning the virtual servers themselves are not directly accessible publicly. Furthermore, to make the servers available for new data uploading and protect the confidentiality and anonymity of the projects using this application, data are deleted every three hours.

4. Example

4.1 Overview of the case study and the corpus

The case study chosen for the experiment of retrieving relevant data and illustration of the Relevance Filter application is the debate on the new cybersecurity law of Vietnamese government. As the law has received sharp criticism from the United States government, the European Union council and human rights groups and the attempts to increase control of the Internet by the authoritarian regimes are on the rise, it is important to study and understand how such law was perceived by civil society, and what arguments the authoritarian regime used to defend and justify their totalitarian mode of control. To obtain the government's narratives and counter narratives, Facebook posts of the following actors from 06/06/2018 (the launching day of the online petition against the cybersecurity law on www.change.org[3]) to 11/10/2018 (the day of the research) were retrieved via Netvizz, an application that collects and extracts data from Facebook (Rieder 2013):

- Three pro-regime Facebook pages that have clear association with the Task Force 47 cyber unit, which operates under the supervision of the Vietnam People's Army to counter perceived "wrong opinions" (Mai 2017, para. 8).
- Three activist groups: Hate Change, Nhật Ký Yêu Nước (translation: Patriotic Diary), and Luật Khoa Tạp Chí (English name: Legal Initiatives for Vietnam)
- One media outlet: BBC Vietnamese

[1] Let's Encrypt is a free, automated, and open certificate authority. See more at https://letsencrypt.org/.
[2] See the petition at https://bit.ly/2sQHD61
[3] See the petition at https://bit.ly/2sQHD61

(See Appendix 1 for more details on the above pages)

The corpora examined in this study consisted of 1,991 posts (or documents) in total. Although they are not large corpora, it was expected that more than half of the posts are irrelevant to the cybersecurity law, especially the posts on BBC Vietnamese. The corpora are likely to comprise of at least three different writing styles by the pro-regime forces, the activists, and the BBC. By choosing these corpora, the experiment aimed to test the hypothesis that each of these groups use different words and phrases in their debate or report on the cybersecurity law. The BBC corpus was included in the experiment to test the rigorousness of the proposed approach for corpus with a large number of irrelevant documents.

A pre-compiled Vietnamese stop-word list developed by Van-Duyet Le (2015) was downloaded from GitHub[4]. As the list is in the text format, I converted it to UTF-8 CSV format. In addition, I added the emerging stopwords as mentioned in the Stopwords section.

4.2 Findings

Comparing the bigrams by the civil society and the pro-regime forces, it becomes clear that while there are some keywords that are used by both, most keywords are distinctively used by only either the civil society or the pro-regime forces. For example, the activist groups use the hashtags savenet, phandoiluatanninhmang, and hoanluatanm in their posts, whereas the pro-regime forces mention cloud computing and the representative offices of Google and Facebook in Vietnam.

Keywords in the activist corpus	Keywords in the pro-regime corpus
An ninh	an ninh
Luật An	ninh mạng

[4] https://github.com/stopwords/vietnamese-stopwords.

mạng xã	lưu trữ
dự thảo	xâm nhập
LUẬT AN	dịch vụ
AN NINH	văn phòng
An ninh	đại diện
ngôn luận	Google Facebook
Internet Việt	máy chủ
mạng Trung	trữ liệu
Dự thảo	an toàn
máy chủ	thông mạng
Việt Nam	viễn thông
ninh mạng	gian mạng
lưu trữ	điện toán
Nam Facebook	đám mây
tài khoản	phạm an
khoản mạng	luật an
dụng Internet	bảo mật
gian mạng	thông não
DỰ LUẬT	công mạng
phản đối	hiểu đúng
Công bố	gian mạng
hoanluatanm savenet	ngôn luận
savenet phandoiluatanninhmang	18 nước
nghệ Huawei	Quốc hội

Figure 3. The keyword lists extracted from the activist corpus and the pro-regime corpus

The evaluation of the outcome of the application shows that it classified 1954 over 1991 documents correctly, which means the precision of the application in this

experiment is around 98%. On the other hand, in contrast with the initial hypothesis, the relevance ranking of the BBC corpus was 100% correct while the result of the biggest corpus of the activist groups is highly questioning. This could be because the number of relevant posts is too small that the relevant keywords did not turn up in the 1,000 most-repeated bigrams.

Facebook Pages (in abbreviation)	Total number of posts	Posts with Relevance Score >=4 (by the application)	Posts with Relevance Score >=4 but irrelevant (after re-evaluation)	Posts with Relevance Score <4 but highly relevant (after re-evaluation)
SAR	124	9	7	0
TF47	237	12	1	1
IAR	146	2	1	0
NKYN	497	9	9	4
Hate change	184	94	0	5
LKTC	214	60	3	6
BBC	589	14	0	0

Figure 4. The Facebook pages and their corresponding statistics after generating the relevance score and re-evaluating the accuracy of the relevance score.

4.3 Discussion

While the result of this experiment is positive, it should be noted that the author was very familiar with the datasets. Nevertheless, the lesson from this experiment is that researchers should read a dataset that includes many relevant data to familiarize themselves with the words and phrases that are likely to be used in other datasets. The relevance ranking would be more efficient if researchers choose highly relevant and distinct keywords. Although the examples used to illustrate the application of the application Relevance Filter are data from Facebook, it should be noted that the

application would be useful to filter textual data from any source given that the corpus are saved in a UTF-8 CSV format.

5. Future Development

Software development is a continuous process to optimize software to meet the needs of users; therefore, the Relevance Filter application will be updated to improve the outcome of matching bigrams and information represented in the documents. As discussed above, the predetermined number of bigrams to be generated might limit the application of the tool to corpus with a large amount of irrelevant data. Therefore, it is envisioned that the next update of the application will allow researchers to set the number of bigrams to be generated from the corpus from 100 to 2,000 bigrams. Users will also be able to download the list of bigrams generated from the corpus since it could be useful for analyzing the linguistic aspects of the corpus. Moreover, the current method mostly accounts for the relevant bigrams in measuring the relevance of the text. It could be improved by taking into account the irrelevant bigrams in the relevance score. To be more specific, irrelevant bigrams, as determined by researchers, would be given deducted score. By doing so, the application will produce better relevance score and it will become more useful for corpus comprising of a large amount of irrelevant data.

6. Conclusion

In this paper, I have described an application to generate a bag-of-bigrams and measure the relevance of textual data in a large corpus comprising of unstructured massive amount of text. The approach of this application is not totally new as it uses the basic method of matching queries and information representation, and the relevance ranking logic of early information retrieval method. The main contribution of this approach and experiment is that keywords used to find and rank relevant documents should be those that are extracted from the corpus itself. Researcher's knowledge of the topic is, nevertheless, critical to selecting the bigrams that are most certainly relevant to the topic and excluding the combination of words that are too

generic or might reduce the efficiency of the relevance score. The procedures of making the stop-word list and selecting keywords require rigorous consideration of information represented in the documents and the context in which these keywords may appear. Another advantage of this approach is that instead of only extracting relevant information and thereby removing the rest of the dataset, it retains the unmatching documents in the dataset; thus, researchers could manually re-evaluate them to avoid missing important documents.

The design of Relevance Filter application is inspired by exchanges with data scientists and researchers from a variety of disciplines and aims to propose a flexible tool that accommodates the retrieval of textual data that are inconsistent or diverse in style and languages.

Acknowledgements

I would like to thank Dr. Bernhard Rieder at University of Amsterdam for his dedicated instructions during the tutorial "Code to Method" and the development of this project. I would also like to thank Mr. Silvan van der Veen for his technical guidance and emotional support throughout all the stages of the project.

Appendix I

Names of the pages	English translations	Abbreviations (defined by the researcher)	URL
Học viện chống phản động	School of anti-reactionary	SAR	https://www.facebook.com/hocvienphongchongphandong/
Lực lượng 47	Task force 47 unit	TF47	https://www.facebook.com/lucluong47/
Trang thông tin chống phản động	Information against reactionaries	IAR	https://www.facebook.com/cpdvn/
Hate Change		Hate Change	https://www.facebook.com/hatechange/
Nhật Ký Yêu Nước	Patriotic Diary	NKYN	https://www.facebook.com/nhatkyyeunuoc1/
Luật Khoa Tạp Chí	Legal Initiatives for Vietnam	LKTC	https://www.facebook.com/luatkhoa.org/
BBC News Tiếng Việt	BBC News Vietnamese	BBC	https://www.facebook.com/BBCVietnamese/

Figure 5. Facebook pages in the experiment

References

Blair, David C., and M. E. Maron. “An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System.” *Communications of the ACM*, vol. 28, no. 3, 1985, pp. 289–299., doi:10.1145/3166.3197.

Chu, Heting. *Information representation and retrieval in the digital age*. Information Today, Inc., 2003.

Fürnkranz, Johannes. "A study using n-gram features for text categorization." *Austrian Research Institute for Artifical Intelligence* 3.1998 (1998): 1-10.

Lo, R. T.-W., He, B., and Ounis, I. (2005). Automatically building a stopword list for an information retrieval system. In Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR).

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development, 1(4):309–317

Makrehchi, M. and Kamel, M. S. (2008). Automatic extraction of domain-specific stopwords from labeled documents. In Advances in information retrieval, pages 222– 233. Springer.

Mai, H. (2017, December 24). Hơn 10.000 người trong 'Lực lượng 47' đấu tranh trên mạng. Retrieved from https://tuoitre.vn/hon-10-000-nguoi-trong-luc-luong-47-dau-tranh-tren-mang-20171225150602912.htm.

Rieder, Bernhard. “Studying Facebook via Data Extraction.” *Proceedings of the 5th Annual ACM Web Science Conference on - WebSci 13*, 2013, doi:10.1145/2464464.2464475.

Sinka, M. P. and Corne, D. (2003a). Evolving better stoplists for document clustering and web intelligence. In HIS, pages 1015–1023

Trumbach, C. C. and Payne, D. (2007). Identifying synonymous concepts in preparation for technology mining. Journal of Information Science, 33(6):660–677.

W3techs. “Usage of UTF-8 for Websites.” *Web Technology Surveys*, https://w3techs.com/technologies/details/en-utf8/all/all.

Yang, Y. (1995). Noise reduction in a statistical approach to text categorization. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pages 256–263. ACM.