ON THE RELEVANCE OF QUERY EXPANSION USING PARALLEL CORPORA AND WORD EMBEDDINGS TO BOOST TEXT DOCUMENT RETRIEVAL PRECISION

Alaidine Ben Ayed[1] and Ismaïl Biskri[2]

[1]Department of Computer Science, Université du Québec à Montréal (UQAM), Canada
[2]Department of Mathematics and Computer Science, Université du Québec à Trois Rivières (UQTR), Canada

ABSTRACT

In this paper we implement a document retrieval system using the Lucene tool and we conduct some experiments in order to compare the efficiency of two different weighting schema: the well-known TF-IDF and the BM25. Then, we expand queries using a comparable corpus (wikipedia) and word embeddings. Obtained results show that the latter method (word embeddings) is a good way to achieve higher precision rates and retrieve more accurate documents.

KEYWORDS

Internet and Web Applications, Data and knowledge Representation, Document Retrieval.

1. INTRODUCTION

Document Retrieval (*DR*) is the process by which a collection of data is represented, stored, and searched for the purpose of knowledge discovery as a response to a user request (query) [1]. Note that with the advent of technology, it became possible to store huge amounts of data. So, the challenge has been always to find out useful document retrieval systems to be used on an everyday basis by a wide variety of users. Thus, *DR* -as a subfield of computer science- has become an important research area. IT is generally concerned by designing different indexing methods and searching techniques. Implementing an *DR* system involves a two-stage process: First, data is represented in a summarized format. This is known as the indexing process. Once, all the data is indexed. users can query the system in order to retrieve relevant information. The first stage takes place off-line. The end user is not directly involved in. The second stage includes filtering, searching, matching and ranking operations.

Query expansion (*QE*) [2] has been a research flied since the early 1960. [3] used *QE* as a technique for literature indexing and searching. [4] incorporated user's feedback to expand the query in order to improve the result of the retrieval process. [5,6] proposed a collection-based term co-occurrence query expansion technique, while [7,8] proposed a cluster-based one. Most of those techniques were tested on a small corpus with short queries and satisfactory result were obtained. Search engines were introduced in1990s. Previously proposed techniques were tested on bigger corpora sizes. We noticed that there was a loss in precision [9,10]. Therefore, *QE* is still a hot search topic, especially in a context of big data.

To measure the accuracy of a *DR* system, there are generally two basic measures [11]: 1) *Precision*: the percentage of relevant retrieved documents and 2) *Recall*: the percentage of documents that are relevant to the query and were in fact retrieved. There is also a standard tool

known as the TRECEVAL tool. It is commonly used by the TREC community for evaluating an ad hoc retrieval run, given the results file and a standard set of judged results.

In this paper we implement a document retrieval system using the Lucene toolkit [12]. Then we investigate the relevance of query expansion using parallel corpora and word embeddings to boost document retrieval precision. The next section describes the proposed system and gives details about the expansion process. The third one describes, and analyses obtained results. The last section concludes this paper and describes the future work.

2. METHODOLOGY

In this section, we present the structure of our Lucene system. First, we describe its core functions. In addition to that we describe some pre-processing operations as well as the evaluation process.

2.1. System Overview

Lucene is a powerful and scalable open source Java-based Search library. It can be easily integrated in any kind of application to add amazing search capabilities to it. It is generally used to index and search any kind of data whether it is structured or not. It provides the core operations for indexing and document searching.

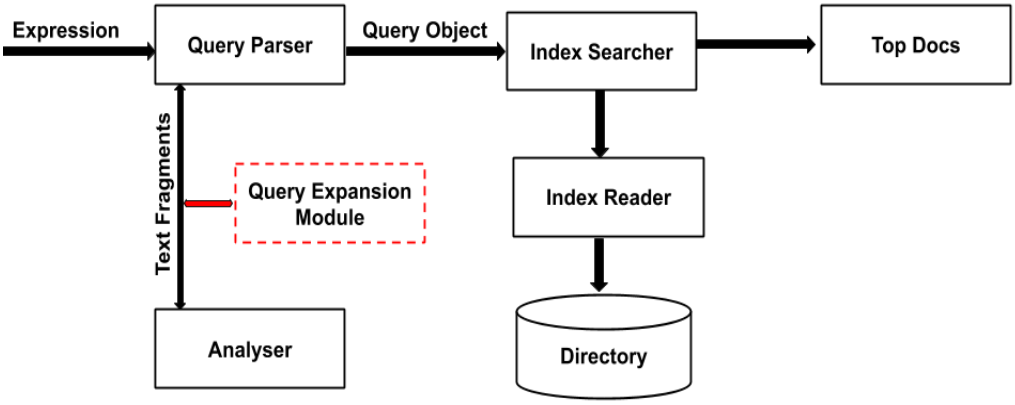

Figure 1: Proposed document retrieval system architecture.

Generally, a Search engine performs all or a few of the following operations illustrated by the above Figure. Implementing it requires performing the following actions:

- Acquire Raw Content: it is the first step. It consists in collecting the target contents used later to be queried in order to retrieve accurate documents.
- Building and analyzing the document: It consists simply in converting raw data to a given format that can be easily understood and interpreted.
- Indexing the document: The goal here is to index documents. So, the retrieval process will be based on certain keys instead of the entire content of the document.

The above operations are performed in an off-line mode. Once all the documents are indexed, users can conduct queries and retrieve documents using the above described system. In this case, an object query is instantiated using a bag of words present in the searched text. Then, the index database is checked to get the relevant details. Returned references are shown to the user. Note

that different weighting schemes can be used in order to index documents. The most used ones are *tfidf* (the reference of the vectoral model) and *BM25* (the reference of the probabilistic model). Typically, the *tf-idf* [13,14] weight is composed by two terms: the first one measures how frequently a term occurs in a document. It computes the normalized Term Frequency (*TF*) which is the ratio of the number of times a word appears in a document by the total number of words in that document. the second term known as the inverse document frequency (*IDF*) measures how important a term is. It computes the ratio of the logarithm of the number of the documents by the number of documents where the specific term appears. *BM25* [15] ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity). It is generally defined as follows:

Given a query Q, containing keywords $q_1, \ldots, q_n$ the *BM25* score of a document D is:

$$score(D,Q) = \sum_{i=1}^{n}(q_i).\frac{f(q_i,D).(k_1+1)}{f(q_i,D)+k_1.(1-b+b.\frac{|D|}{avgdl})}$$

where $f(q_i, D)$ is q_i 's term frequency in the document D, D is the length of the document D in words, and *avgdl* is the average document length in the text collection from which documents are drawn. K_1 and b are free parameters, usually chosen, in absence of an advanced optimization, as $k_1 \in [1.2, 2.0]$ and $b = 0.75$. $IDF(q_i)$ is the *IDF* (inverse document frequency) weight of the query term q_i . It is usually computed as:

$$IDF(q_i) = Log\frac{N-n(q_i)+0.5}{n(q_i)+0.5}$$

where N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i.

2.2. Query Expansion Using a Comparable Corpora and Word Embeddings

In order to improve system accuracy, we proposed two techniques of query expansion. The first one uses Wikipedia as comparable corpus. The second one uses word embeddings. The main purpose is to make the query more informative while reserving its integrity.

2.2.1. Query expansion using a comparable corpus

First, we use Wikipedia as a comparable corpus to expand short queries. For this purpose, we tested two slightly different approaches.

- *Query expansion by summary*: We extract key-words from the query using the Rake algorithm [16]; a domain-independent method for automatically extracting keywords. we rank keywords based on their order of importance, we take the most important one. Then, we use it to query Wikipedia. We summarize the first returned page; AKA, we make a short summary of one sentence and we concatenate it to the original query.
- *Query expansion by content*: We extract key-words from the query using the Rake algorithm. we rank keywords based on their order of importance. Then, we take the most important one and we use it to query Wikipedia. Therefore, we concatenate titles of the top returned pages to the original query.

2.2.2. Query expansion using word embeddings

Word embeddings are also used to expand the queries. We assume that the concept expressed by a given word can be strengthen by adding to the query the bag of words that usually co-occur with it. For this purpose, we use the Gensim implementation of word2vec using three different models: glove-twitter-25, glove-twitter-200, fasttext-wiki-news-subwords-300 and glove-wiki-gigaword-300 [17].

3. EXPERIMENTS, RESULTS AND DISCUSSION

3.1. The Data Set

For experiments, we used a subset of the Trec dataset. It is a news corpus. It consists in a collection of 248500 journal article covering many domains such as economics, politics, science and technology, etc. First, we perform pre-processing of our corpus by removing stop words, applying stemming or lemmatization. Stemming is the process of transforming to the root word by removing common endings. Most common widely used stemming algorithms are Porter, Lancaster and Snowball. The latter one has been used in this project. In lemmatization, context and part of speech are used to determine the inflected form of the word and applies different rules for each part of speech to get the root word (lemma). Obtained results using different pre-processing strategies are reported in the next section.

The most frequently and important basic measures for document retrieval effectiveness are precision and recall [18]. Precision is simply the probability given that an item is retrieved it will be relevant and recall is the probability given that an item is relevant it will be retrieved. In this work, we use the TRECEVAL program [19] to evaluate the proposed system. It uses the mentioned above NIST evaluation procedures.

3.2. Results and Discussion

Obtained results are reported in Table 1, Table 2, Table 3 and Table 4.

Table 1: The importance of pre-processing

Data Type	**original data**			**stemmed data**		
Metric	P5	P10	Map	P5	P10	Map
Short queries	0.192	0.026	0.115	0.196	0.030	0.148
Long queries	0.194	0.030	0.139	0.236	0.037	0.148

Table 2: TFIDF VS. BM25

weighting schema	**TFIDF**			**BM25**		
Metric	P5	P10	Map	P5	P10	Map
Short queries	0.196	0.172	0.148	0.211	0.180	0.152
Long queries	0.236	0.266	0.148	0.242	0.221	0.161

Table 1 shows the system accuracy when using non-processed Vs. pre-processed data. It proves that pre-processing helps to achieve better precision rates. While, Table 2 shows results when using different weighting schema: better results are obtained by using the BM25 weighting schema. Notice here that we performed the same pre-processing before conducting experiences using different weighting schema.

Table 3: Obtained results when expanding queries by summary and content

Expansion Strategy	***O***			***T***			***S***		
Metric	P5	P10	Map	P5	P10	Map	P5	P10	map
Short queries	0.196	0.172	0.148	0.195	0.156	0.149	0.072	0.109	0.057

Table 3 displays obtained results when applying query expansion using a comparable corpus. We conducted two experiences: T (using expanded queries by title), S (expanded queries by topic) and we compared their results to those obtained by our original set of short queries. Notice here that we performed the same pre-processing and we used the same weighting schema. Obtained results show that expanding queries through titles of the top returned Wikipedia pages gives approximately the same precision rates with the original procedure. Whereas, expanding queries through the summary of the Wikipedia top page messes up the system accuracy.

Table 4: Obtained results when expanding queries through word embeddings

Model	**O**			***WE1***			***WE2***			***WE3***			***WE4***		
Metric	P5	P1000	Map	P5	P1000	Map	P5	P1000	map	P5	P1000	Map	P5	P1000	map
Short queries	0.196	0.030	0.148	0.164	0.026	0.018	0.01188	0.027	0.116	0.204	0.032	0.125	0.216	0.028	0.132

Finally, Table 4 shows results when expanding queries using word embeddings. we used the Gensim implementation of word2vec. We tested three different models: glove-twitter-25 (*WE1*), glove-twitter-200 (*WE2)*, fasttext-wiki-news-subwords-300 (*WE3*) and glove-wiki-gigaword-300 (*WE4*). Obtained results show that the system accuracy can be enhanced when taking in consideration the top 5 returned results. To achieve this goal, we should use the appropriate model: WE3 which is trained with a news corpus or *WE4* which is trained using very a huge corpus.

4. CONCLUSIONS AND FUTURE WORK

In this work, a *DR* system based on the Lucene toolkit is presented. Different weighting schema are tested. Results show that the probabilistic model (*BM25*) performs the vectoral one (*TFIDF*). Also, lead experiments show that query expansion using word embeddings improves the overall system precision. Meanwhile, using a comparable corpus doesn't necessarily lead to the same result. This paper can be improved by:

- Testing an interactive query expansion technique: experimental results show that the query expansion using a comparable corpus does not lead to higher precision rates. Actually, the precision rate depends on the efficiency of the Rake key word extractor algorithm. The main idea is to let users validate the automatically extracted keywords used later on during the query expansion process.

- Testing an hybrid technique of query expansion: word embeddings can be applied on the result of the interactive query expansion phase. This may boost the system performance since the interactive query expansion will guarantee the use of significant words of the query. Also, using word embeddings will ensure retrieving relevant documents which do not necessarily contain words used in the query.

Currently, we are adding a new functionality to our system; We are implementing a multi-document text summarization technique which generates a comprehensive summary of the retrieved set of documents.

ACKNOWLEDGEMENTS

The authors would like to thank Natural Sciences and Engineering Research Council of Canada for financing this work.

REFERENCES

[1] Anwar A. Alhenshiri, Web Information Retrieval and Search Engines Techniques,2010, Al-Satil journal, PP:55-92

[2] H. K. Azad, A. D., Query Expansion Techniques for Information Retrieval: a Survey. 2017.

[3] Maron, M.E., Kuhns, J.L.: On relevance, probabilistic indexing and information retrieval. Journal of the ACM(JACM) 7(3), 216-244, 1960.

[4] Rocchio, J.J.: Relevance feedback in information retrieval, 1971.

[5] Jones, K.S.: Automatic keyword classi cation for information retrieval, 197.

[6] van Rijsbergen, C.J.: A theoretical basis for the use of co-occurrence data in information retrieval. Journal ofdocumentation 33(2), 106-119, 1977.

[7] Jardine, N., van Rijsbergen, C.J.: The use of hierarchic clustering in information retrieval. Information storageand retrieval 7(5), 217-240, 1971.

[8] Minker, J., Wilson, G.A., Zimmerman, B.H.: An evaluation of query expansion by the addition of clusteredterms for a document retrieval system. Information Storage and Retrieval 8(6), 329-348, 1972.

[9] Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the AmericanSociety for Information Science 41, 288-297, 1990.

[10] Harman, D.: Relevance feedback and other query modi cation techniques, 1992.

[11] R. Sagayam, S.Srinivasan, S. Roshni, A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques, IJCER, sep 2012, Vol. 2 Issue. 5, PP: 1443-1444.

[12] https://lucene.apache.org/core/

[13] Breitinger, C.; Gipp, B.; Langer, S. Research-paper recommender systems: a literature survey. International Journal on Digital Libraries.2015. 17 (4): 305338.

[14] Hiemstra, Djoerd. A probabilistic justification for using tf×idf term weighting in information retrieval. International Journal on Digital Libraries 3.2 (2000): 131-139.

[15] Stephen E. R.; Steve W.; Susan J.; Micheline H-B. & Mike G. Okapi at TREC3. Proceedings of the Third Text Retrieval Conference (TREC 1994). Gaithersburg, USA.

[16] Stuart J-R, Wendy E-C. Vernon L-C. And Nicholas O-c, Rapid Automatic Keyword Extraction for Information Retrieval and Analysis, G06F17/30616 Selection or weighting of terms for indexing, USA, 2009

[17] Jeffrey P., Richard S., Christopher D-M, GloVe: Global Vectors for Word Representation.

[18] David M.W (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". Journal of Machine Learning Technologies. 2 (1): 3763.

[19] https://trec.nist.gov/

AUTHORS

Alaidine Ben Ayed is a PhD. candidate in cognitive computer science at Université du Québec à Montréal (UQAM), Canada. His research focuses on artificial intelligence, natural language processing (Text summarization and conceptual analysis) and information retrieval.

Ismaïl Biskri is a professor in the Department of Mathematics and Computer Science at the Université du Québec à Trois-Rivières (UQTR), Canada. His research focuses mainly on artificial intelligence, computational linguistics, combinatory logic, natural language processing and information retrieval