International Journal of Information Retrieval Research
Volume 12 • Issue 1

An End-to-End Efficient Lucene-Based Framework of Document/ Information Retrieval

Alaidine Ben Ayed, Université du Québec à Montréal, Canada
Ismaïl Biskri, Université du Québec à Trois-Rivières, Canada
Jean-Guy Meunier, Université du Québec à Montréal, Canada

ABSTRACT

In the context of big data and the Industrial Revolution 4.0 era, enhancing document/information retrieval framework efficiency to handle the ever-growing volume of text data in an ever more digital world is a must. This article describes a double-stage system of document/information retrieval. First, a Lucene-based document retrieval tool is implemented, and a couple of query expansion techniques using a comparable corpus (Wikipedia) and word embeddings are proposed and tested. Second, a retention-fidelity summarization protocol is performed on top of the retrieved documents to create a short, accurate, and fluent extract of a longer retrieved single document (or a set of top retrieved documents). Obtained results show that using word embeddings is an excellent way to achieve higher precision rates and retrieve more accurate documents. Also, obtained summaries satisfy the retention and fidelity criteria of relevant summaries.

KEYWORDS

Data and Knowledge Representation, Document Retrieval, Internet and Web Applications, Mono/Multi-Document Summarization

INTRODUCTION

Document Retrieval (*DR*) is defined as the process of matching some stated user queries against a set of free-text records (Anwar, 2010). Nowadays, Massive and quite variant data is being generated at an unprecedented rate. In this context, the big data era has overturned classical *DR* challenges. More focus is being addressed on proposing innovative indexing and searching routines. Document retrieval systems generally perform two basic operations: *1)* indexing; is the process of representing data in a condensed format, *2)* querying; is the process of querying the *DR* system to retrieve appropriate data. The first operation does not involve end-users. Generally, it is performed in an off-line mode. The second one includes numerous processing operations, ranging from filtering, searching, mapping to ranking returned indexes.

Document retrieval frameworks are built upon the cluster hypothesis (Fiana & Oren, 2013). Identifying the appropriate cluster of pertinent documents to a given straightforward user query is an easy task. Finding the set of clusters appropriate to complex queries is a more difficult task (Tombros et al., 2002) (Liu & Croft, 2006). The retrieval performance drops down if top accurate documents are not presented at the top of returned indexes. Proposing new ranking query-specific cluster strategies has been a hot research topic for many years (Leuski, 2001), and suggested solutions

DOI: 10.4018/IJIRR.289950

base on a cluster-against-query representation comparison (Liu & Croft, 2004) (Liu & Croft, 2008). Some document retrieval frameworks make use of extra features, including inter-cluster and cluster-document similarities (Kurland & Lee, 2006) (Kurland & Domshlak, 2008) (Kurland & Krikon, 2011). Query expansion (*QE*) is another way to heighten document retrieval systems accuracy (Hiteshwar & Akshay, 2019).

First attempts of query expansion have been proposed since early 1960. The main objective is to improve the retrieval process performance. In this context, *QE* was used as a procedure for literature indexing and searching (Maron & Kuhns, 1960). The user's feedback was employed in (Rocchio, 1971) to expand queries. (Jones, 1971) (Van, 1977) suggested a collection-based term co-occurrence *QE* protocol, while (Jardine & Van, 1971) (Minker, 1972) introduced a cluster-based one. The mentioned above techniques led to satisfactory results. Nevertheless, they were experimented with using small corpora and a set of straightforward user queries. Researchers noticed a considerable loss in retrieval precision when the mentioned above techniques were tested using bigger corpora sizes provided by public search engines, firstly implemented in 1990 (Salton & Buckley, 1990) (Harman, 1992). Consequently, query expansion has been a hot search topic, notably in an ever-growing big data word. *Precision* and *Recall* are the states of the art standard measures of document retrieval accuracy (Sagayam et al., 2012). The first one refers to the percentage of relevant retrieved records, while the second one refers to the percentage of relevant records being retrieved. Notice also that the document retrieval research community uses *TRECEVAL*[1], a standard tool commonly used to evaluate ad hoc retrieval runs, given the returned documents and a conventional collection of refereed results.

Automatic text summarization (*ATS*) is another critical research area related to text document retrieval if we assume that the returned result may be a concise, reliable, and fluid extract of a given longer retrieved text document. *ATS* can also be applied to a set of retrieved documents. Generally, automatic text summarization is either performed by extraction (Mehdi et al., 2017) (Andhale & Bewoor, 2016) or abstraction (Yogan et al., 2016). The first approach extracts prominent sentences that vehicle the essential concepts of the source text. Nevertheless, the latter creates novel sentences by applying rephrasing techniques instead of merely reporting the most salient fragments.

Note that extractive models gained more attention than abstractive approaches. Extractive summarizers generally estimate a relevancy score for each sentence of the original retrieved source text document. The latter score determines to what extent a given sentence encodes significant concepts. The generated output is made of the top-scored sentences. This kind of summarization remains a challenging research field. The extraction process depends on a set of linguistic and-or statistical features.

Linguistic-based summarizers tend to build a formal representation of the conveyed information through the text to summarize. The central intuition behind it is to employ discourse analysis techniques to model text rhetoric. For instance, the Rhetorical Structure Analysis (*RST*) can be employed to find out "*Nucleus*" fragments, which contain salient information, and "*satellite*" ones delivering additional information about the *nucleus*. In this specific case, "*Nucleus*" sentences would have higher relevancy scores, and they will be chosen to be part of the generated summary (Barzilay & Elhadad, 1999) (Kundi et al., 2014).

On the other hand, there are three variants of statistically-based summarizers; *i)* frequency-based, *ii)* feature-based, and *iii)* machine-learning-based ones. Frequency-based summarizers are built upon one of two primary hypotheses. The first one is *a)*: "cue words would be repeated many times in a given text document"; in this case, term frequency (Nenkova et al., 2006) (Nenkova & Vanderwende, 2005) is used to compute sentence relevancy scores. Inverse document frequency (a kind of probabilistic measure) (Filatova & Hatzivassiloglou, 2004) (Fung & Ngai, 2006) (Galley, 2006) is used for the same reason if we assume *b)*: "essential words are more frequent in a given document than in another one.". In other words, the inverse document frequency measure estimates how relevant a word is to a given text in a set or text records. Feature-based statistical summarizers employ a bunch of indicators to compute sentence relevancy scores. Those indicators are mainly; the

presence of cue headline tokens, sentence length or position, etc. (Gupta & Lehal, 2010). Machine learning-based summarizers makes use of training data to learn "*relevant*" and "*non-relevant*" sentence patterns (Svore et al., 2007) (Burges et al, 2005) (Hannah & Mukherjee, 2014).

Automatically generated summaries are regarded as straightforward, reliable, and fluid abstracts if they meet three main criteria:

- ***Retention*:** It is a measure of the extent to which the generated summary covers different topics discussed in the retrieved document (or set of documents).
- ***Fidelity*:** It is a measure of the extent to which the summary accurately reflects the author's point of view(s).
- ***Coherence*:** It is a measure of the extent to which the generated extract is semantically meaningful.

This article presents a *Lucene*-based document retrieval framework. It proposes two query expansion techniques: The first one uses parallel corpora while the second one bases on word embeddings to boost retrieval accuracy. Next, it adds to the proposed *DR* framework, a mono/multi-document summarization layer. A concise, reliable, and fluid extract of a given longer retrieved text document (or a set of documents) is returned as a query result instead of a crud index (or a set of indexes). The coming section describes the suggested document retrieval framework and details the proposed expansion and summarization protocols. The third one reports obtained results. The last section concludes this article and details ongoing and planned work.

METHODOLOGY

This section presents the suggested *Lucene*-based *DR* framework as well as the proposed query expansion techniques. Also, it describes the theoretical details of the summarization process.

System Overview

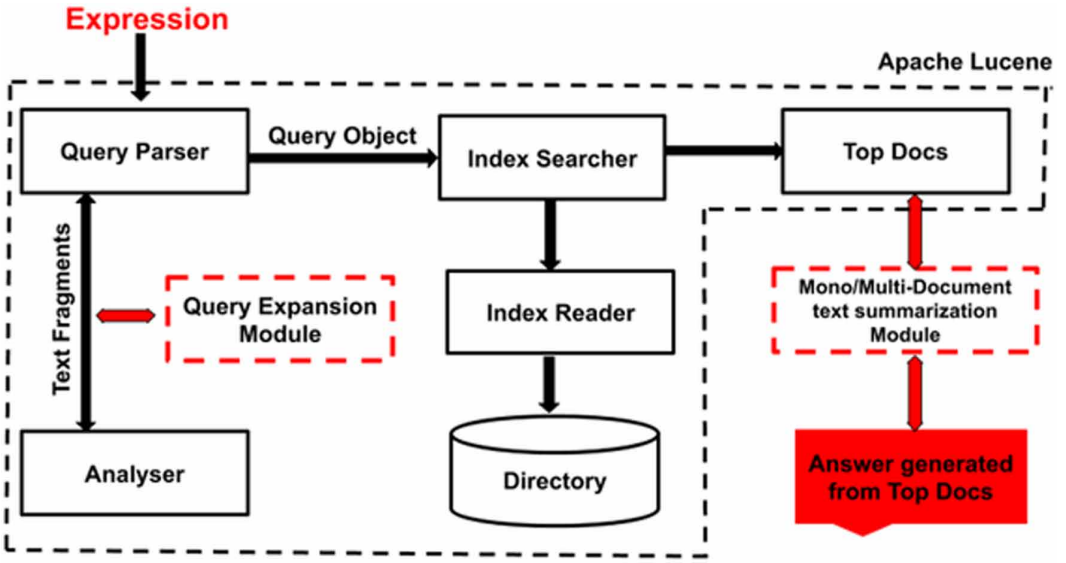
Lucene[2] is a robust and scalable open-source Java-based Search library. It can be easily integrated into any application to add impressive search capabilities to it. It implements the core services needed for indexing and record searching for both structured and no structured data.

The proposed *IR/DR* framework performs the following processes shown by the above Illustration:

1. Acquiring crude contents: this first step refers to collecting the data utilized later to be queried during the retrieval phase.
2. Analyzing crude documents: It consists of merely converting each instance of the crude data to a given format that can be efficiently guessed and rendered.
3. Indexing data: It consists of mapping each document by a specific key. Next, the retrieval process will base on particular keys rather than the entire document's full content.
4. Retrieving top documents: It consists of returning indexes of top matching documents to the user query.
5. Summarizing a given retrieved document or a set of documents: It consists of returning an abstract of the top retrieved document (or a set of the top retrieved documents).

Operations *1)*, *2)*, and *3)* are generally performed off-line. Users can query the described *DR* framework; after that, all the text records are appropriately indexed. Generally, indexing documents bases either on a vectorial model (*TF-IDF*) or a probabilistic one (*BM25*). The query is converted onto a bag of words, and the index database is investigated to get the query response. Any returned reference is displayed to the user as a concise, reliable, and fluid extract of a given longer retrieved text document.

Figure 1. The proposed document retrieval framework architecture



TF-IDF estimates how relevant a word *w* is to a document *d* in a corpus of text documents *D*. It is computed by multiplying two different quantities (Breitinger et al., 2015) (Hiemstra, 2000):

- **The term frequency (*tf*):** refers to the number of occurrences of *w* in *d*. Usually, the term frequency is adjusted by the *d* 's length or *d* 's most recurrent word frequency.
- ***The inverse document frequency (idf):*** refers to how common or rare *w* is in the entire corpus *D*. Being close to 0 means that the *w* is commonly used in *D*.

The higher the *TF-IDF* score is, the more relevant that word is in that particular document. The *TF-IDF* score for the word *w* in *d*; a document belonging to a set of documents *D* is computed as follows:

$$TFIDF\left(w,d,D\right) = tf\left(w,d\right) .idf\left(w,D\right) \tag{1}$$

where: $tf\left(w,d\right)=log\left(1+freq\left(w,d\right)\right)$ and $idf\left(w,D\right)=log\left(\frac{N}{count\left(d\in D:w\in d\right)}\right)$.

BM25 (Stephen & Karen, 1994) ignores the inter-relationship between the query terms within a document. Its ranking process works as follows: Given a query Q, containing canonic words q_1, ..., q_n, the *BM25* score of a document *d* is defined as:

$$score\left(d,Q\right) = \sum_{i=1}^{n}\left(q_i\right)\frac{f\left(q_i,N\right).\left(k_1+1\right)}{f\left(q_i,N\right)+k_1\left(1-b+b\frac{\left|N\right|}{avgdl}\right)} \tag{2}$$

where $f(q_i, d)$ refers to the q_i 's term frequency in the document d, N refers to the document d 's length, and *avgdl* is the average length of all text documents. K_1 and b are free parameters, usually empirically fixed as $k_1 \in [1.2, 2.0]$ and $b = 0.75$.

Query Expansion to Boost Retrieval Accuracy

The next couple sub-sections describe the proposed two query expansion techniques. The goal is to boost the proposed document retrieval framework accuracy by making user queries more informative while preserving their integrity.

Comparable Corpora-Based Query Expansion

The first proposed technique of query expansion uses *Wikipedia* as a comparable corpus. Two slightly different variants of the same approach are described below:

- ***Summary-based query expansion*:** The *Rake* algorithm (Stuart et al., 2009) is used to extract keywords. *Rake* is a domain-independent keyword extraction technique. It returns a list of keywords in a text with their order of importance. The most important keyword is used as a canonic word to query *Wikipedia*. Following this, a one-sentence summary of the first returned *Wikipedia* page is generated. Next, it is concatenated to the original query.
- ***Content-based query expansion*:** the most crucial keyword returned by the *RAKE* algorithm is used to query *Wikipedia*. Next, the top returned *Wikipedia* page's title is concatenated to the user's query to make it more informative.

Word Embeddings-Based Query Expansion

The main idea is to expand any user query by terms having the closest embedding representation to its relevant terms. For instance, if the query contains the word "bumper," it is expanded by a set of semantically close words like "brackets," "fillers," and "parts" since "bumper," "bumper brackets," "bumper fillers" and "car parts" usually co-occur together. In this way, all text records or technical sheets related to "bumper brackets" would be considered when searching for relevant records for "bumper" even though the word "brackets" is not present in the user query. The *Gensim* implementation of *word2vec* is used to find relevant expanding terms to the user query. In this context, three distinct models were tested, namely *fasttext-wiki-news-subwords-300, glove-twitter-25, glove-twitter-200*, and *glove-wiki-gigaword-300* (Jeffrey et al, 2014).

Automatic Document Summarization

Automatic text summarization is used as a top layer of the proposed Lucene based document retrieval system. It abstracts a single retrieved document (or a set of retrieved documents) to create a short, accurate, and fluent extract. In the case of multi-retrieved document summarization, all retrieved texts are concatenated and considered a single retrieved document. Mathematically, the main idea is to project the document to summarize onto a lower-dimensional space that captures the essence of concepts present in the source text. The latter space's unitary vectors are used to compute *retention-fidelity* scores, as described in our paper (Alaidine et al., 2019). The mathematical and implementation details of the proposed summarization protocol will be expanded in the coming two subsections.

Retention-Fidelity (RF) Tensor Construction

First, a lexicon, including all unique non-generic words, is constructed. Next, each retrieved text is segmented into n sentences. Each sentence S_i is represented by a sentence column feature vector x_i. x_i is a vector of d components. Note that d is equal to the lexicon's cardinality. Each component of x_i represents the number of occurrences of a given word of the lexicon in the text to summarize.

$$x_i = \begin{pmatrix} w_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{pmatrix} \text{ and } x_i^T = \left(x_{i1}, x_{i2}, \ldots, x_{id}\right) \tag{3}$$

A set of sentence feature vectors is strongly correlated if one or many components are simultaneously highly activated. For instance, if the number of occurrences of one or more tokens like "economy," "system," "private," "individuals," "businesses," "own-capital" exceeds a given threshold, it would be probable that this set of sentences are discussing the concept of "Capitalism." The goal is to project the crude sentence feature dataset from many correlated coordinates onto fewer uncorrelated ones called principal concepts while still retaining most of the original data's variability. Thus, sentence feature vectors are stacked as rows of a data matrix to construct the crude text feature matrix.

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{pmatrix} \tag{4}$$

The mean sentence vector (equation 5) is subtracted from each sentence feature vector to remove noise and redundant information (equation 6). Next, the normalized text feature matrix is constructed by stacking zero-centered sentence feature vectors as its rows.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \left(\frac{1}{n}\sum_{i=1}^{n} x_{i1} \ , \ \ldots, \ \frac{1}{n}\sum_{i=1}^{n} x_{id}\right)^T \tag{5}$$

$$X = \begin{pmatrix} x_{1-\mu^T}^T \\ x_{2-\mu^T}^T \\ \vdots \\ x_{n-\mu^T}^T \end{pmatrix} \tag{6}$$

Next, the covariance around the mean is computed as follows:

$$S = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \mu\right)\left(x_i - \mu\right)^T = \frac{1}{n-1} X^T X \tag{7}$$

As said bellow, the motivation is to project the crude sentence feature vectors dataset from many correlated coordinates onto fewer uncorrelated ones called principal concepts. Vectors encoding those concepts will be built sequentially in a way that maximizes their contributions to the variances of the

original set of sentence feature vectors. Mathematically, the goal is to find a collection of $k \leq d$ unit vectors $vi \in \mathbb{R}^d$ (for $i \in 1,\ldots,k$) called principal concepts, such that:

1. The variance of the set of sentence feature vectors projected onto the v_i direction is maximized.
2. v_i should be orthogonal to $v_1,\ldots,v_{i-1}$.

The projection of a vector $x \in \mathbb{R}^d$ onto the line determined by any v_i is simply given as the dot product $v_i^T x$. The variance of the sentence feature vector x projected onto the first principal concept v_1 is defined as follows:

$$S = \frac{1}{n-1}\sum_{i=1}^{n}\left(v_1^T x_i - v_1^T \mu\right)^2 = v_1^T S v_1 \tag{8}$$

To construct v_1, S is maximized while satisfying the $v_1 \|=1$ additional constraint. The Lagrange multipliers (*LM*) approach is used to solve this optimization problem. *LM* implies that $Sv_1 = \lambda_1 v_1$,aka; v_1 is an eigen concept (mathematically, it is an eigenvector of the covariance matrix S). Note that $\| v_1 \|= v_1^T v_1 = 1$, this means that the corresponding eigenvalue is equal to $v_1^T S v_1 = \lambda_1$. It equals the variance of the sentence feature vectors along v_1 . The most important concept is coded by the eigenvector associated to the highest eigenvalue.

Next, the sentence feature vectors set is projected onto a new direction v_2 , the same way, while satisfying the $v_1 \perp v_2$ condition, then onto v_3 while satisfying $v_3 \perp v_1, v_2$, and so on.

By the end of this process, the first k vectors encoding principal concepts of X are built. They are eigenvectors of the covariance matrix S corresponding to its k highest eigenvalues. Next, the conceptual space will be constructed such that the k most important eigen concepts will form its orthonormal basis $\check{\mathrm{z}}_k$:

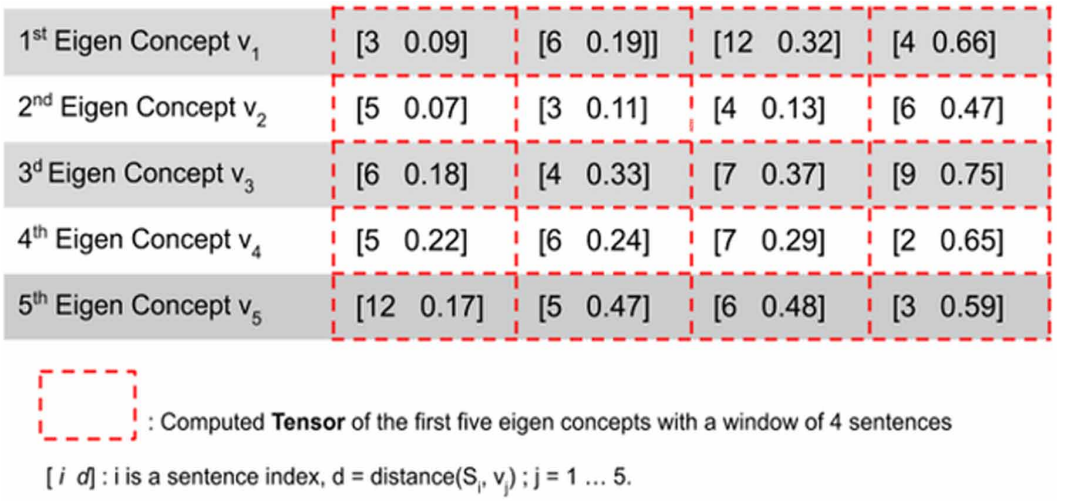
$$\Xi_k = \left[v_1, v_2, \ldots, v_k\right] \tag{9}$$

Each normalized projected sentence onto the constructed conceptual space can be written as a linear combination of k eigen concepts. Next, the goal is to build a retention-fidelity tensor. Thus, the Euclidean distance between a given concept v_j ; $j = 1,\ldots,k$ and any normalized sentence $\widehat{x_i} = x_i - \mu$, projected in the conceptual space is defined and computed as follows:

$$d_i\left(v_j\right) = v_j - \widehat{x_i} \tag{10}$$

The *Retention-Fidelity* tensor provides distances between algebraic sentence feature vectors and the orthonormal conceptual space basis's unitary vectors. It is constructed such that the line order depends on the importance of a given concept, while the column order is related to the extent to which a random sentence encodes a given concept. For instance, the first line provides the w best sentences to encode the first most crucial concept (their normalized projected feature vectors have

Figure 2. Retention-Fidelity tensor construction using the five most important eigen concepts and a window size w=4



the smallest distances to v_1 encoding the most important concept). The second line provides the same information related to the second most important concept, and so on. Note also that the fifth sentence, for instance, is the best sentence to encode the second most crucial concept, while the sixth sentence is the last one in a window size of four sentences.

As described in the coming section, the Retention-Fidelity tensor will be used to compute a Retention-Fidelity score for each sentence.

Retention-Fidelity (RF) Score Computation and Summary Construction

First, a *Retention* score is computed for each normalized sentence being projected onto the constructed conceptual space. A given sentence having a high *Retention* sore should encode as much as possible the most important concepts expressed in the retrieved text document. In other words, it should appear as much as possible in a window of size w while taking into consideration the k important concepts. Mathematically, it is defined as follows:

$$R_{kw}(s) = \frac{1}{k}\sum_{i=1}^{k}\alpha_i \tag{11}$$

$\alpha_i = 1$ if the sentence S occurs in the i^{th} window. If not, it is equal to zero.

Now, an extended fidelity ($F_{kw}(s)$) score is computed for every sentence. It is a kind of averaged sum of the *retention* coefficient. The latter one is weighted according to the sentence's position in each window of size w. The central intuition is that sentences with a high F_{kw} score should encode important concepts while focusing on the most important ones. The *fidelity* score is defined as follows:

$$F_{kw}(s) = \frac{1}{k}\sum_{i=1}^{k}\alpha_i\left[1+\frac{1-\psi_i}{w}\right] \tag{12}$$

$\alpha_i = 1$ if a sentence s occurs in the i^{th} window. If not, it is equal to zero. ψ_i is the rank of a sentence s in the i^{th} window.

Next, Fuzzy logic is used to compute a unified *Retention-Fidelity* (*RF*) score for every sentence of the retrieved document following the previously described protocol in (Alaidine et al., 2019). Highly scored sentences are extracted to generate a concise abstract as a response to the user's query.

EXPERIMENTS, RESULTS AND DISCUSSION

The Data Set

The *Trec* dataset, a news corpus of 248500 journal articles, is used for experiments. It covers many fields such as politics, economics, technology, science, etc. Crude data is preprocessed by removing stop words and applying stemming routines. The stemming technique consists of removing common endings to transform words to their root form. The most common widely used stemming algorithms are Porter, Lancaster, and Snowball. The latter is used in this project.

Precision and *recall* are commonly used to measure document retrieval effectiveness (David, 2011). *Precision* refers to the probability given that a text is retrieved; it will be relevant. *Recall* refers to the probability given that a text is relevant; it will be retrieved. In this research paper, the *TRECEVAL* program is used to evaluate the retrieval accuracy. It uses the mentioned below evaluation procedures:

- **P5:** Precision after 5 docs retrieved.
- **P100:** Precision after 1000 docs retrieved.
- **MAP:** Mean Average Precision.

Also, the *FRESA* protocol (Juan-Manuel et al., 2010) is used to evaluate the quality of the generated summaries.

Results and Discussion

Query expansion effectiveness related results are reported in Tables 1, 2, 3, and 4.

Table 1 compares system accuracy when using non-processed VS. pre-processed data. It confirms that pre-processing helps to reach better accuracy rates. Table 2 compares the obtained retrieval precisions using two different weighting schemas (*TF-IDF* and *BM25*). Note that the same pre-processing protocol was used in both scenarios. Generally, the *BM25* weighting schema outperforms the *TF-IDF* one.

Table 1. On the relevance of pre-processing to improve document retrieval accuracy

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**

Obtained results when using a comparable corpus as an expansion technique are reported in Table 3. Three main experiences were conducted: *O)*; precision without any expansion technique, *C)*; expanding user queries by content and *S)*; expanding user queries by summaries. Note that the same pre-processing was performed and, the same weighting schema is used for experiments *O)*, *C)*, and

Table 2. Document retrieval accuracy when using TF-IDF VS. BM25 weighting schemas

Weighting Schema	TF-IDF			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 3. Obtained results when expanding queries by summary and content

Expansion Strategy	O			C			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 4. Obtained results when expanding queries using word embeddings

Expansion Strategy	O			WE1			WE2			WE3			WE4		
Metric	P5	P10	Map	P5	P10	Map	P5	P10	Map	P5	P10	Map	P5	P10	Map
Short queries	0.196	0.030	0.148	0.164	0.026	0.018	0.011	0.027	0.116	**0.204**	0.032	0.125	**0.216**	0.028	0.132

S). Reported results in Table 3 show that using titles of the top returned Wikipedia pages to expand user queries provides almost the same accuracy rates as without using any query expansion technique. Using the summary of the Wikipedia top page to expand user queries messes up the retrieval precision.

Table 4 reports obtained results when using word embeddings to expand user queries. Glove-twitter-25 (*WE1*), glove-twitter-200 (*WE2*), fasttext-wiki-news-subwords-300 (*WE3*), and glove-wiki-gigaword-300 (*WE4*) word2vec variants provided by The *Gensim* implementation of *word2vec* are used to perform the expansion process. Achieved results confirm that the system retrieval accuracy can be improved when considering the top 5 retried documents under a critical constraint that consists of choosing the appropriate *word2vec* model. In the bellow example, *WE3*, which is trained using a collection of news articles, and *WE4*, which is trained using a massive corpus of textual data, helped ameliorate retrieval accuracy. It was not the case when using the inappropriate *word2vec* model to this specific context.

The *FREZA* evaluation protocol (Juan-Manuel et al., 2010) was used to evaluate the quality of the generated abstracts (Table 5). The best results are obtained when we summarize the first retrieved document. For multi-document summarization, the best results are obtained when we summarize

Table 5. Obtained Fresa scores for mono/multi document summarization with window sizes w = 2,4 and 6

Window size (w)	Mono-Document Summarization	Multi-Document summarization ($D = 2$)	Multi-Document summarization ($D = 5$)	Multi-Document summarization ($D = 10$)
$w = 2$	0.642	0.590	0.398	0.136
$w = 4$	**0.748**	**0.711**	0.473	0.213
$w = 6$	0.219	0.340	**0.591**	0.311

fewer retrieved documents ($D = 2$) with a window size $w = 4$. If we want to summarize more than two retrieved documents (the top five ones, for instance) while approximately preserving the same quality of the returned result, we should consider a bigger window size ($w = 6$). Generally, summarizing many documents ($D = 10$) deteriorates the quality of the final query response.

CONCLUSION AND FUTURE WORK

This paper presented a *Lucene* based document retrieval framework. Comparing two different weighting schemas (*TF-IDF* and *BM25*) shows that the *BM25* probabilistic model outperforms the vectorial one (*TF-IDF*). Additionally, led query expansion experiments show that using word embeddings enhances the overall document retrieval precision. It is not the case when using a comparable corpus. The proposed framework can be enhanced by implementing an interactive query expansion approach: The obtained result using the proposed comparable corpus-based expansion approach depends on the efficiency of the *Rake* keyword extractor algorithm. The central intuition is to involve users in the query expansion process. Users have to approve the returned extracted keywords. Another hybrid technique of query expansion may be explored: Once the user validates keywords, a word embeddings expansion will be performed. The latter technique will ensure using only appropriate keywords and retrieving relevant records that do not necessarily contain terms used in the user query.

Mono-document and multi-document summarization of the few top retrieved references achieved excellent coverage and fidelity levels, fundamental criteria of useful summaries. Note that the coherence of the generated query response is out of the scope of this paper. Applying a discourse analysis technique like the Rhetorical structure theory (*RST*) establishes a formal representation of the retrieved document's knowledge. It helps to generate more coherent abstracts (Mann & Thompson, 1988). Achieving a fully coherent abstract of mono-document summaries is straightforward. Employing the rhetorical structure theory technique, as mentioned above, can quickly achieve it. For multi-document summarization, local coherence can be achieved, and the main challenge would be to achieve a global coherence. Finally, the proposed summarization protocol can be improved to create a more professional human-like abstract of the top retrieved documents; Rephrasing techniques can be performed on extracted text segments to generate original sentences instead of merely extracting the most salient ones.

ACKNOWLEDGMENT

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

REFERENCES

Alaidine, B. A., Biskri, I., & Jean-Guy, M. (2019). Automatic Text Summarization: A New Hybrid Model Based on Vector Space Modelling, Fuzzy Logic and Rhetorical Structure Analysis. In *Computational Collective Intelligence 2019* (pp. 26–34). Springer International Publishing.

Andhale, N., & Bewoor, L. A. (2016). An overview of Text Summarization techniques. *Proceedings of the International Conference on Computing Communication Control and automation (ICCUBEA)*. 1–7.

Anwar, A. A. (2010). Web Information Retrieval and Search Engines Techniques. *Journal Al-Satil*, 55-92.

Barzilay, R., & Elhadad, M. (1999). Using lexical chains for text summarization. In *Advances in Automatic Text Summarization* (pp. 111–121). The MIT Press.

Breitinger, C., Gipp, B., & Langer, S. (2015). Research-paper recommender systems: A literature survey. *International Journal on Digital Libraries*, *17*(4), 305–338.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., & Deeds, M. (2005). Learning to rank using gradient descent. *Proceedings of the International Conference on Machin eLearning (CML' 05/ACM)*, 89–96.

David, M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Fiana, R., & Oren, K. (2013). Ranking Document Clusters Using Markov Random Fields. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*.

Filatova, E., & Hatzivassiloglou, V. (2004). A formal model for information selection in multi-sentence text extraction. *Proceedings of the 20th International Conference on Computational Linguistics*, 397–403.

Fung, P., & Ngai, G. (2006). One story, one flow: Hidden Markov story models for multi-lingual multi-document summarization. *ACM Transactions, Speech Language Processing*, 1–16.

Galley, M. (2006). A skip-chain conditional random field for ranking meeting utterances by importance. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (NLP' 06)*, 364–372.

Gupta, V., & Lehal, G. S. (2010). A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, *2*(3), 258–268.

Hannah, M. E., & Mukherjee, S. (2014). A classification-based summarization model for summarizing text documents. *International Journal of Information and Communication Technology*, *6*(3/4), 292–308.

Harman, D. (1992). Relevance feedback and other query modification techniques. *Journal of Information Retrieval: Data Structures and Algorithms*, 241-263.

Hiemstra, D. (2000). A probabilistic justification for using *tf-idf* term weighting in information retrieval. *International Journal on Digital Libraries*, *3*(2), 131–139.

Hiteshwar, K. Z., & Akshay, D. (2019). Query Expansion Techniques for Information Retrieval: A Survey. *Journal of Information Processing and Management*.

Jardine, N. & Van, R. (1971). The use of hierarchic clustering in information retrieval. *Journal of Information Storage and Retrieval*, *7*(5), 217-240.

Jeffrey, P., Richard, S., & Christopher, D. M. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Jones, K. S. (1971). *Automatic keyword classification for information retrieval*. Archon Books.

Juan-Manuel, T. M., Horacio, S., Iria, D. C., & Eric, S. (2010). Summary Evaluation with and Without References. *Journal of Polibits*, *42*(42), 13–20.

Kundi, F. M., Ahmad, S., Khan, A., & Asghar, M. Z. (2014). Detection and Scoring of Internet Slangs for Sentiment Analysis Using SentiWordNet. *Journal of Life Science*, 66–72.

Kurland, O., & Krikon, E. (2011). The opposite of smoothing: A language model approach to ranking query-specific document clusters. *Journal of Artificial Intelligence Research*, *41*, 367–395. doi:10.1613/jair.3327

Kurland, O. & Domshlak, C. (2008). A rank-aggregation approach to searching for optimal query-specific clusters. *Proceedings of SIGIR*, 547–554.

Kurland, O. & Lee, L. (2006). Respect my authority! HITS without hyperlinks, utilizing cluster-based language models. *Proceedings of SIGIR*, 83–90.

Leuski, A. (2001). Evaluating document clustering for interactive information retrieval. *Proceedings of the tenth international conference on Information and knowledge management*, 33–40. doi:10.1145/502585.502592

Liu, X., & Croft, W. B. (2004). Cluster-based retrieval using language models. *Proceedings of SIGIR*, 186–193.

Liu, X., & Croft, W. B. (2006). *Experiments on retrieval of optimal clusters.* Technical Report IR-478, Canter for Intelligent Information Retrieval (CIIR), University of Massachusetts.

Liu, X., & Croft, W. B. (2008). Evaluating text representations for retrieval of the best group of documents. *Proceedings of ECIR*, 454–462.

Mann, W. C., & Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a functional theory of text organization. *Journal of Text & Talk*, *8*(3), 243–281.

Maron, M. E., & Kuhns, J. L. (1960). On relevance, probabilistic indexing, and information retrieval. *Journal of the Association for Computing Machinery*, *7*(3), 216–244.

Mehdi, A., Seyedamin, P., Mehdi, A., Saeid, S., Elizabeth, D. T., Juan, B. G., & Krys, K. (2017). Text Summarization Techniques: A Brief Survey. *International Journal of Advanced Computer Science and Applications*, *8*(10).

Minker, J., Wilson, G. A., & Zimmerman, B. H. (1972). An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Journal of Information Storage and Retrieval*, *8*(6), 329–348.

Nenkova, A., & Vanderwende, L. (2005). *The impact of frequency on summarization*. Microsoft Research.

Nenkova, A., Vanderwende, L., & McKeown, K. (2006). A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. *29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (5), 573–580.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. *Proceedings of SIGIR*.

Sagayam, R., Srinivasan, S., & Roshni, S. (2012). A survey of text mining: Retrieval, extraction and indexing techniques. *International Journal of Computational Engineering Research*, *2*(5), 1443–1446.

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, *41*, 288–297.

Stephen, E. R. & Karen, S. J. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, *27*(3), 129–146.

Stuart, J. R., Wendy, E. C. Vernon, L. C., & Nicholas, O. C. (2009). *Rapid Automatic Keyword Extraction for Information Retrieval and Analysis.* US Pents: G06F17/30616: Selection or weighting of terms for indexing.

Svore, K. M., Vanderwende, L., & Burges, C. J. (2007). Enhancing single-document summarization by combining RankNet and third-party sources. *Microsoft Research.*

Tombros, A., Villa, R., & Van, R. C. (2002). The effectiveness of query-specific hierarchic clustering in information retrieval. *Journal of Information Processing and Management*, *38*(4), 559–582. doi:10.1016/S0306-4573(01)00048-6

Van, R. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *The Journal of Documentation*, *33*(2), 106–119.

Yogan, J. K., Ong, S. G., Halizah, B., Ngo, H. C., & Puspalata, C. S. (2016). A Review on Automatic Text Summarization Approaches. *Journal of Computational Science*, *12*(4), 178–190.

ENDNOTES

1 https://trec.nist.gov/
2 https://lucene.apache.org/core/

Alaidine Ben Ayed is a Ph.D. candidate in Cognitive Computer Science at Université du Québec à Montréal (UQAM), Canada. His research mainly focuses on cognitive artificial intelligence, natural language processing (text summarization and conceptual analysis), and information retrieval.

Ismaïl Biskri is full professor in computational linguistics and artificial intelligence at the computer science department of the University of Quebec at Trois-Rivières. He is the head of the laboratory in applied artificial intelligence. His research interests concern aspects of fundamental research on the syntactic and functional semantic analysis of natural languages with using models of Categorial Grammars and combinatory logic. He also works on specific issues in text-mining, classification, information retrieval, and terminology. His research is funded by the Canadian granting agencies FQRSC, SSHRC, and NSERC.