Front. Comput. Sci., 2018, 12(1): 163–176
https://doi.org/10.1007/s11704-016-5560-0

RESEARCH ARTICLE

Strength Pareto fitness assignment for pseudo-relevance feedback: application to MEDLINE

Ilyes KHENNAK (✉), Habiba DRIAS

Laboratory for Research in Artificial Intelligence, Computer Science Department, University of Sciences and Technology Houari Boumediene (USTHB), Algiers 16111, Algeria

Abstract Because of users' growing utilization of unclear and imprecise keywords when characterizing their information need, it has become necessary to expand their original search queries with additional words that best capture their actual intent. The selection of the terms that are suitable for use as additional words is in general dependent on the degree of relatedness between each candidate expansion term and the query keywords. In this paper, we propose two criteria for evaluating the degree of relatedness between a candidate expansion word and the query keywords: (1) co-occurrence frequency, where more importance is attributed to terms occurring in the largest possible number of documents where the query keywords appear; (2) proximity, where more importance is assigned to terms having a short distance from the query terms within documents. We also employ the strength Pareto fitness assignment in order to satisfy both criteria simultaneously. The results of our numerical experiments on MEDLINE, the online medical information database, show that the proposed approach significantly enhances the retrieval performance as compared to the baseline.

Keywords information retrieval, query expansion, pseudo-relevance feedback, proximity, multi-objective optimization, Pareto dominance, MEDLINE

1 Introduction

Both the amount of data available on the World Wide Web (WWW) and the number of newly created Web pages are continuously increasing. Ranganathan [1] showed in his research that the volume of online data indexed by Google increased from 5 EB in 2002 to 280 EB in 2009. According to Zhu et al. [2], this volume is expected to double every 18 months. Ntoulas et al. [3] interpreted these findings in terms of the number of new Web pages created and estimated that their number is growing by 8% a week. In their study, Bharat and Broder [4] went further and demonstrated that the number of Web pages is increasing at the rate of 7.5 every second. The unprecedented explosion of information available on the WWW has led to the following results.

- New keywords are continuously being invented and introduced into the WWW. Williams and Zobel [5] showed that one of every two hundred keywords used is new. Studies by [5–7] indicated that this massive influx is largely due to the first occurrences of rare personal names and place names, neologisms, acronyms, abbreviations, emoticons, typographical errors, and URLs.
- WWW users are constantly exploiting these new keywords in their search queries. In their study, Chen et al. [8] indicated that more than 17% of query terms are not dictionary words, i.e., out of vocabulary, 45% of them are E-speak expressions, 18% are products and companies, 16% are proper names, and 15% are misspellings and foreign words [9, 10].

The difficulty of disambiguating the sense of these new unclear and imprecise keywords has caused search engines systems to fail to find the desired information. One of the most

Received December 26, 2015; accepted September 12, 2016

E-mail: heoui@live.com

powerful and effective methods to overcome this shortcoming is query expansion (QE). This method aims to augment the original search queries with the additional keywords that best characterize the users' needs [11]. Various QE approaches addressing the proximity and interdependence of words have been developed and tested. They predominantly focused on examining and assessing the degree of relatedness between an additional keyword candidate and the user search query in order to select the most appropriate keywords that can be added to the initial search query. Despite its high effectiveness as compared to previous methods, the performance of QE has still not reached a sufficiently effective level to allow its use as a standard component.

Accordingly, in order to improve the performance of QE, we propose in this paper a new approach for expanding an original user query with additional terms that best express the actual user intent. We introduce two criteria to assess the degree of relatedness between a candidate expansion term and the search query keywords: (1) co-occurrence, where the good Turing discounting (GTD) method is used to attribute more importance to words that occur in the largest possible number of documents where the search query keywords appear; (2) proximity, where the Kernel functions is used to assign more importance to words at a short distance from the query terms within the documents. Furthermore, we adopt the strength Pareto fitness assignment to satisfy both criteria simultaneously.

We extensively evaluate the proposed approach using the MEDLINE database, the world's largest medical library. In addition, we use the two well-known pseudo-relevance feedback (PRF) techniques, Rocchio's method and the Robertson and Sparck Jones (RSJ) term-ranking function combined with Okapi BM25, as the baseline for comparison. We also compare the proposed approach with the Robertson Selection Value (RSV), Kullback-Leibler Distance (KLD), and Idealized Relevance Feedback (IRF) methods.

The main contributions of this paper are as follows.

- The adoption of an external correlation measure based on GTD to evaluate the co-occurrence of terms with respect to the search query keywords.
- The determination of an internal correlation measure based on kernel functions to assess the proximity of words with respect to the search query keywords.
- The use of strength Pareto fitness assignment to satisfy both correlation measures simultaneously.

The remainder of this paper is organized as follows. In the following section, we shortly review the state-of-the-art QE methods and present some concepts and definitions, covering PRF and Okapi BM25. In Section 3, we present the proposed Pareto dominance based on GTD and kernel functions to select the best expansion keywords. Experimental and numerical results are given in Section 4 and we conclude the paper in Section 5.

2 Related work

The massive influx of new terms on the WWW, such as first occurrences of proper names, abbreviations, and misspelled words, as well as the use of these unclear and ambiguous terms to describe the user's information need, have caused the failure of search engines to retrieve the relevant information. In addition, the term mismatch problem and the vocabulary problem still remain the most serious issues currently confronting the retrieval effectiveness of search engines. To handle these serious problems, various methods have been proposed, including interactive query refinement (Google suggest), word sense disambiguation [12], search result clustering [13], Boolean term decomposition [14], spreading activation networks [15], concept lattice-based information retrieval [16], random indexing [17], and contextual document ranking modeled as basis vectors [18]. Nevertheless, the expansion of the user's search query with additional terms is one of the most powerful and effective methods to improve the retrieval effectiveness of document ranking [11]. Currently, QE is widely used in numerous applications, such as question answering [19, 20], cross-language information retrieval [21], multimedia information retrieval [22, 23], information filtering [24], text categorization [25], search of hidden Web content that is not indexed by standard search engines [26], query completion on mobile devices [27], training corpora acquisition [28], e-commerce [29], mobile search [30], expert finding [31], slot-based document retrieval [32], federated search [33], and paid search advertising [34].

The process of selecting the most relevant and related terms to be used as expansion keywords is the key step in QE. Several concepts, such as proximity, co-occurrence, association, closeness, relatedness, and relationship, have been introduced and discussed in order to describe the strength of the correlation between an expansion term candidate and the search query keywords. The extraction of semantic relationships between terms has been extensively adopted in QE through the use of dictionaries and thesauruses, such as WordNet. The work of Voorhees [35] was among the first in which WordNet was used as a tool for QE and exploited to enrich the queries using a combination of synonyms, hypernyms, and hyponyms. Collins-Thompson and Callan [36] employed WordNet to combine multiple sources of knowledge on term associations through a Markov chain framework. The authors used the stationary distribution of the model to obtain probability estimates that a candidate expansion term reflects aspects of the original query. In the same direction, Liu et al. [37] adopted WordNet as a background thesaurus and source of expansion candidates. They performed phrase recognition and sophisticated word sense disambiguation on queries, and then selected highly-correlated terms having the same sense as the query terms. Song et al. [38] proposed a semantic QE technique that combines association rules with WordNet and natural language processing techniques, utilizes the explicit semantics, as well as other linguistic properties of unstructured text corpus, and utilizes the contextual properties of important terms discovered by association rules to select the appropriate expansion keywords.

The detection of similarities between keywords has also been explored by examining the contents of a database in order to capture relationships between terms and build an association thesaurus by utilizing context vectors [39], mutual information [40], latent semantic indexing [41], and interlinked Wikipedia articles [42].

Another class of techniques involves extracting the most favorable keywords from the top ranked documents retrieved by the original search query for use as expansion terms. For instance, Rocchio [43] and Robertson et al. [44] took the terms that are initially returned for the search query and used information about whether or not the terms are relevant to reformulate the original query. Similarly, Wong et al. [45] attempted to automatically formulate effective queries using full or partial relevance information in the context of relevance feedback.

In recent papers, the creation of statistical language models by determining a probability distribution over terms was proposed. Zhai and Lafferty [46] proposed a QE method based on statistical language models, namely, the divergence minimization model. It used an idea similar to the Rocchio algorithm [43] and selected the words with the highest probabilities for expansion. An additional related work is that of Lavrenko and Croft [47]. They explored the relation between classical probabilistic models of information retrieval and the emerging language modeling approaches to generate the best expansion keywords.

With the aim of expanding the original search query with related terms, we propose in the present paper an effective approach that uses both co-occurrence and proximity criteria to assess the strength of the relationship between a candidate expansion term and the search query keywords. The novelty of our approach as compared to previous methods lies in the adoption of the strength Pareto fitness assignment to fulfill simultaneously the above two criteria. A preliminary version of the proposed approach was highlighted in our earlier paper, which was presented at the 3rd World Conference on Information Systems and Technologies in 2015 [48].

2.1 Strength Pareto fitness assignment for generating expansion keywords

As mentioned above, the current work was briefly described in our previous paper [48]. In that study, we sought to show that measuring the strength of the relationship between potential candidate terms and the query terms on the basis of their positions within the documents and their distribution in the top ranked documents can facilitate the selection of the best expansion terms. Furthermore, to measure the degree of relatedness between terms, we attempted to adopt both the GTD method and kernel functions. We also attempted to demonstrate that adjusting the balance between GTD and kernel functions using Pareto dominance may achieve better results.

The major difference between the previous and present contributions is that the earlier one provided preliminary ideas and suggestions that were insufficiently detailed and had not been tested. In contrast, the current paper describes the proposed approach in detail, as well as its validation by extensive experiments. Furthermore, the present study investigated for the first time how to effectively adopt and combine the co-occurrence, the proximity, and the Pareto dominance with a PRF technique. Moreover, unlike the previous one, this paper discusses prior work and reviews some preliminaries and concepts, covering PRF, Pareto dominance, GTD, and kernel functions.

2.2 Pseudo-relevance feedback for query expansion

The expansion of the original search query with additional terms that best capture the actual user intent is one of the most natural and successful techniques to improve the retrieval effectiveness of document ranking. Many approaches have been introduced to generate and extract these additional terms. PRF, also called Retrieval Feedback, is one of the proposed approaches. It extracts the most appropriate terms to be used as expansion keywords from the pseudo-relevant documents, i.e., the first documents returned in response to the original search query.

In its simplest version, PRF first runs an initial search on the original search query using the best instantiation of the probabilistic relevance framework, Okapi BM25 (see Section 2.2.1). Then, it extracts the expansion keyword candidates from the pseudo-relevant documents and ranks them using a term-scoring function (see Section 2.2.2). Finally, it augments the initial search query with the best expansion keywords and retrieves the documents relevant to the expanded query.

2.2.1 Probabilistic relevance framework, Okapi BM25

The probabilistic relevance framework is a framework for document retrieval that led to the development of one of the most powerful text retrieval algorithms, Okapi BM25. The classic version of the Okapi BM25 term-weighting function, in which the weight w_i^{BM25} is attributed to a given term t_i in a document d, is given by

$$w_i^{BM25} = \frac{tf}{k_1\left((1-b) + b\dfrac{dl}{avdl}\right) + tf} w_i^{RSJ}, \tag{1}$$

where tf is the frequency of the term t_i in document d, k_1 and b are constants, dl is the document length, and $avdl$ is the average document length. w_i^{RSJ} is the well-known RSJ weight [44] and is calculated as

$$w_i^{RSJ} = \log\frac{(r_i + 0.5)(N - R - n_i + r_i + 0.5)}{(n_i - r_i + 0.5)(R - r_i + 0.5)}, \tag{2}$$

where N is the number of documents in the whole collection, n_i is the number of documents in the whole collection that contain t_i, R is the number of documents judged to be relevant, and r_i is the number of documents that are judged to be relevant and contain t_i.

The RSJ weight can be used with or without relevance information. In the absence of relevance information, this weight is reduced to a form of classical IDF:

$$w_i^{IDF} = \log\frac{N - n_i + 0.5}{n_i + 0.5}. \tag{3}$$

The final Okapi BM25 term-weighting function is then given by

$$w_i^{BM25} = \frac{tf}{k_1\left((1-b) + b\dfrac{dl}{avdl}\right) + tf} \log\frac{N - n_i + 0.5}{n_i + 0.5}. \tag{4}$$

A large number of experiments were conducted to determine the parameters k_1 and b. The results of these experiments showed that values such as $1.2 < k_1 < 2.0$ and $0.5 < b < 0.8$ are reasonably good in many cases. In the same context, Robertson and Zaragoza [49] demonstrated in their study that recent versions of Okapi BM25 are based on specific values assigned to k_1 and b: $k_1 = 2.0$, $b = 0.5$.

As part of the indexing process, an inverted index is generated including the weight w_i^{BM25} of each term t_i in each document d. The similarity score between the document d and a given query q is then calculated using the Okapi BM25 document-scoring function:

$$Score_{BM25}(d, q) = \sum_{t_i \in q} w_i^{BM25}. \tag{5}$$

During the search process, the relevant documents are retrieved and ranked using the similarity score mentioned above.

2.2.2 Term-scoring functions for pseudo-relevance feedback

As reported earlier, the PRF method first runs a preliminary search on the initial query using Okapi BM25 term-weighting and previous document scoring functions (Eqs. (4) and (5), respectively), supposing the top ranked documents to be relevant, attributing a score to each term in the top ranked documents using a term-scoring function and then sorting them on the basis of their scores. One well-known term-scoring function is RSJ, defined by Eq. (2). Other term-scoring functions are as follows.

Rocchio's weight [43]:

$$w_i^{Rocchio} = \sum_{d \in R} w_i^{BM25}; \tag{6}$$

RSV [50]:

$$w_i^{RSV} = \sum_{d \in R} w_i^{BM25} \times (r_i + 0.5); \tag{7}$$

KLD [51]:

$$w_i^{KLD} = (r_i + 0.5)\log\left(\frac{r_i + 0.5}{R - r_i + 0.5}\right); \tag{8}$$

IRF [45]:

$$w_i^{IRF} = \alpha \sum_{d \in R} w_i^{RSJ} - \gamma \sum_{d \notin R} w_i^{RSJ}, \tag{9}$$

where R is the set of pseudo-relevant documents and α and γ are constants. In their study, Wong et al. [45] set α and γ to 0.9 and 0.4, respectively.

The original search query is finally augmented by adding the top ranked terms and re-interrogated using the Okapi BM25 document-scoring function to obtain more relevant results.

In this study, the Rocchio weight, as well as the RSJ, RSV, KLD, and IRF term-scoring functions, were used as the baseline for comparison.

3 Pareto dominance for selecting expansion keywords

The main goal of the proposed approach is to improve the retrieval effectiveness and return only the documents that are relevant to the search query. For this purpose, in this study, we used the concepts of co-occurrence and proximity to extract the best expansion keywords to be added to the original search query. These concepts are based first on finding for each query term q_i the locations and positions where it appears and then selecting from the locations the candidate terms that frequently neighbor and co-occur with that query term. In other words, we recover for each query term q_i the documents where it appears, and then assess the relevance of the candidate terms contained in these documents with respect to the query term q_i on the basis of:

- The co-occurrence, which gives a value to candidate words that appear in the largest possible number of those documents.
- The proximity, which gives a value to candidate words in which the distance separating them and the query term q_i within a given document, in terms of the number of words, is small.

These candidate words are then sorted on the basis of their relevance to the whole query and the top ranked ones are added to that query to repeat the search process and obtain more relevant results.

Before proceeding to describe the concepts of co-occurrence and proximity, we first need to represent each candidate term $t_i \in V_R$ by a vector T_i of $|R|$ elements, as follows:

$$T_i = \langle pos_1, pos_2, \ldots, pos_{|R|} \rangle, \tag{10}$$

where R is the set of pseudo-relevant documents returned by Eq. (5), V_R is the vocabulary of R, and pos_k is the position(s) of t_i in the pseudo-relevant document d_k. In the case where $t_i \notin d_k$, the value of pos_k is 0; otherwise, its value is a vector containing all possible positions of t_i in d_k.

As indicated earlier, as a first step we find the candidate words that frequently appear together with the query keywords. These candidate words are found by attributing more value to terms that occur in the largest possible number of documents where each of the query keywords appears. We interpret this value through the measurement of the external correlation ext of each term $t_i \in V_R$ to each term $t_{j(q)}$ of the query q. This correlation, which does not take into consideration the content of documents, computes the rate of appearance of t_i with $t_{j(q)}$ in the set of documents R. The external correlation of t_i to $t_{j(q)}$ is significant when t_i appears in the largest number of documents in which $t_{j(q)}$ occurs, and vice versa. Based on this interpretation, the external correlation of t_i to $t_{j(q)}$ is calculated using the GTD method as

$$ext\left(t_i, t_{j(q)}\right) = \frac{1}{C(t_{j(q)})}\left[\left(C(t_i, t_{j(q)}) + 1\right)\frac{N_{C+1}}{N_C}\right], \tag{11}$$

where $C(t_{j(q)})$ is the number of times that $T_{j(q)}[k] \neq 0$, where $k = 1, 2, \ldots, |R|$ (i.e., the number of documents where $t_{j(q)}$ occurs in R). $C(t_i, t_{j(q)})$ is the number of times that $(T_{j(q)}[k], T_i[k]) \neq 0$, where $k = 1, 2, \ldots, |R|$ (i.e., the number of documents where t_i and $t_{j(q)}$ co-occur in R). N_{C+1} is the number of pairs of terms that include $t_{j(q)}$ and occur $C+1$ times in R, and N_C is the number of pairs of terms that include $t_{j(q)}$ and occur C times in R.

The GTD method has been widely used for computing the probability of a complete string of words or providing a probabilistic prediction of the next word in a sentence. In practice, GTD has been utilized to assign a non-zero probability to sequences of N words (N-grams) with zero or low counts by examining the number of N-grams with higher counts [52]. Our dependence on GTD is the result of our need to solve the issue that we faced in our previous studies. In those studies, we attempted to adopt the classical conditional probability to compute the rate of appearance of a given term relative to another one. The main problem related to using the conditional probability was that words originally having a low occurrence frequency were neglected. Thus, their overall rates of appearance were automatically decreased. If words with a low occurrence frequency are ignored, this implies the words that were indicated earlier (i.e., first occurrences of rare personal names and place names, abbreviations, acronyms, etc.) are omitted. Accordingly, to avoid dropping the candidate words with low frequency, we use GTD to re-estimate their low probabilities and improve their low appearance rates.

After we have computed the rate of appearance of t_i with each query term $t_{j(q)}$, the overall external correlation between t_i and the whole query q is represented by an array containing all possible ext between t_i and each query term $t_{j(q)}$:

$$ext(t_i, q) = \langle ext(t_i, t_{1(q)}), ext(t_i, t_{2(q)}), \ldots, ext(t_i, t_{|q|(q)}) \rangle. \tag{12}$$

The cosine similarity measure is then used to evaluate the quality of each vector $ext(t_i, q)$ with respect to the best vector $ext(t_i^*, q)$, where each of its elements $ext(t_i^*, t_{j(q)})$ represents the highest external correlation between the term t_i and $t_{j(q)}$. The following function, $f_{ext}(t_i)$, indicates the cosine similarity score between $ext(t_i^*, q)$ and $ext(t_i, q)$:

$$f_{ext}(t_i) = \frac{\sum_{j=1}^{|q|} ext(t_i, t_{j(q)}) \times ext(t_i^*, t_{j(q)})}{\sqrt{\sum_{j=1}^{|q|} \left[ext(t_i, t_{j(q)})\right]^2} \times \sqrt{\sum_{j=1}^{|q|} \left[ext(t_i^*, t_{j(q)})\right]^2}}. \tag{13}$$

In the second step, we attempt to find the candidate words that are frequently neighbors of the query keywords. Therefore, we assign more value to words in close proximity to the query keywords. We interpret this importance through the measurement of the internal correlation between each term t_i of V_R and each term $t_{j(q)}$ of the query q. This correlation computes the correlation between t_i and $t_{j(q)}$ within a given document d_k in terms of the number of words separating them. The closer t_i is to $t_{j(q)}$ within d_k, the greater is its internal correlation. We use the well-known kernel functions to measure the internal correlation:

Gaussian kernel:

$$K(i, j) = \exp\left[\frac{-(i-j)^2}{2\sigma^2}\right]; \tag{14}$$

Triangle kernel:

$$K(i, j) = \begin{cases} 1 - \dfrac{i-j}{\sigma}, & \text{if } |i-j| <= \sigma; \\ 0, & \text{otherwise}; \end{cases} \tag{15}$$

Cosine kernel:

$$K(i, j) = \begin{cases} \dfrac{1}{2}\left[1 + \cos\left(\dfrac{|i-j|\pi}{\sigma}\right)\right], & \text{if } |i-j| <= \sigma; \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

where σ is a parameter to be tuned.

Using the kernel functions, the internal correlation *int* between t_i and $t_{j(q)}$ within a given document d_k is then calculated by

$$int(t_i, t_{j(q)})_{d_k} = K\left(T_i[k], T_{j(q)}[k]\right). \tag{17}$$

Next, the average internal correlation between t_i and $t_{j(q)}$ in the whole R is determined as

$$int(t_i, t_{j(q)}) = \frac{1}{C(t_{j(q)})} \sum_{d_k \in R} int(t_i, t_{j(q)})_{d_k}. \tag{18}$$

The overall internal correlation between t_i and the whole query q is described by a vector containing all possible *int* between t_i and each query term $t_{j(q)}$:

$$int(t_i, q) = \langle int(t_i, t_{1(q)}), int(t_i, t_{2(q)}), \ldots, int(t_i, t_{|q|(q)}) \rangle. \tag{19}$$

The cosine similarity measure is then used to assess the quality of each array $int(t_i, q)$ with respect to the best vector $int(t_i^*, q)$, where each of its elements $int(t_i^*, t_{j(q)})$ represents the highest internal correlation between a given term t_i and $t_{j(q)}$. The following function, $f_{int}(t_i)$, indicates the cosine similarity score between $int(t_i^*, q)$ and $int(t_i, q)$:

$$f_{int}(t_i) = \frac{\sum_{j=1}^{|q|} int(t_i, t_{j(q)}) \times int(t_i^*, t_{j(q)})}{\sqrt{\sum_{j=1}^{|q|} \left[int(t_i, t_{j(q)})\right]^2} \times \sqrt{\sum_{j=1}^{|q|} \left[int(t_i^*, t_{j(q)})\right]^2}}. \tag{20}$$

Finally, in order to extract the terms that are suitable for use as expansion keywords, we adopt the well-known concept of Pareto dominance. Thus, instead of using the conventional methods when determining the overall correlation, such as summing the internal and external correlations or adjusting the balance between them, we consider both the internal and the external correlations as multiple conflict criteria to be fulfilled simultaneously. The concept of Pareto dominance was proposed in order to solve the multi-objective optimization problem, also called multi-criteria optimization. The multi-objective optimization problem can be defined as the problem of finding a solution that satisfies an objective vector, the elements of which represent the objective functions. The solution to this problem can be described in terms of a decision vector $(x_1, x_2, \ldots, x_n)$ in the decision space X. A function $f : X \to Y$ evaluates the quality of a given solution by assigning to it an objective vector $(y_1, y_2, \ldots, y_k)$ in the objective space Y. We say that a decision vector x^1 is better than another decision vector $x^2 (x^1 > x^2)$ if the objective vector y^1 dominates the objective vector $y^2 (y^1 > y^2)$, where $y^1 = f(x^1)$, $y^2 = f(x^2)$, and $k > 1$. The vector y^1 is said to dominate the vector y^2 if no component of y^1 is smaller than the corresponding component of y^2, and at least one component of y^1 is greater than the corresponding component of y^2. The set of optimal solutions, i.e., solutions not dominated by any other solutions, in the decision space X is denoted as the Pareto set $X^* \subseteq X$ and its image in objective space is denoted as the Pareto front $Y^* = f(X^*) \subseteq Y$. Many enhanced functions have been proposed and extended to describe the Pareto dominance concept. One of the most improved dominance functions is the strength Pareto fitness assignment (SPEA2). It assigns to each solution x^i a strength value $S(x^i)$ representing the number of solutions it dominates:

$$S(x^i) = |x^j | x^j \in X \wedge x^i > x^j|, \tag{21}$$

where |.| is the cardinality of set, > is the Pareto dominance relation, and $x^i > x^j$, if the objective vector y^i assigned to x^i dominates the objective vector y^j assigned to x^j.

On the basis of the S values, the raw fitness $R(x^i)$ of solution x^i is calculated by

$$R(x^i) = \sum_{x^j \in X, x^j > x^i} S(x^j). \tag{22}$$

It is important to note that the raw fitness R is to be minimized here; i.e., $R(x^i) = 0$ corresponds to a non-dominated individual.

By analogy, for the proposed approach, a candidate expansion term $t_i \in V_R$ is represented by a decision vector $T_i = \langle pos_1, pos_2, \ldots, pos_{|R|} \rangle$. The quality of a given candidate term t_i is evaluated by assigning it an objective vector $f(t_i) = \langle f_{ext}(t_i), f_{int}(t_i) \rangle$. We say that a candidate term t_i is better than another candidate term t_j ($t_i > t_j$) if the objective vector $f(t_i)$ dominates the objective vector $f(t_j)$. The vector $f(t_i)$ is said to dominate the vector $f(t_j)$ if its components, $f_{ext}(t_i)$ and $f_{int}(t_i)$, are not smaller than their corresponding components in $f(t_j)$, and at least one component of $f(t_i)$ is greater than its corresponding component in $f(t_j)$. Based on the concept of dominance, each candidate term t_i is assigned a strength value $S(t_i)$ representing the number of expansion keyword candidates it dominates:

$$S(t_i) = |t_j | t_j \in V_R \wedge t_i > t_j|. \tag{23}$$

The raw fitness $R(t_i)$ of each candidate term $t_i \in V_R$ is then calculated by

$$R(t_i) = \sum_{t_j \in V_R, t_j > t_i} S(t_j). \tag{24}$$

Next, the candidate terms are sorted on the basis of their raw fitness values. The best candidate terms are those with the lowest raw fitness values and the top ranked ones are added to the original search query q.

Based on the Okapi BM25 document-scoring function, presented in Section 2, the relevant documents are retrieved using

$$Score_{BM25}(d, \grave{q}) = \sum_{t_i \in q} w_i^{BM25} + \frac{1}{2} \sum_{t_i \in \grave{q} - q} w_i^{BM25} \times [f_{ext}(t_i) + f_{int}(t_i)], \tag{25}$$

where $\grave{q}$ is the expanded query.

4 Experiments

In order to assess the quality of the proposed approach, we conducted a set of experiments. First, we describe the dataset, the software, and the effectiveness measures used. Then, we present the experimental results.

4.1 Dataset

The experiments were performed on the MEDLINE database, the world's largest medical library. This database includes 348, 566 references consisting of titles and/or abstracts from 270 medical journals over a five-year period (1987–1991). The available fields are title, abstract, MeSH indexing terms, author, source, and publication type. In addition, the collection contains a set of queries and relevance judgments (a list of which documents are relevant to each query).

In order to obtain convincing and credible results, we divided the MEDLINE dataset into six sub-collections. Each sub-collection is defined by a set of documents, queries, and a list of relevance documents. Table 1 highlights the characteristics of each sub-collection in terms of the number of documents it contains (docs), the size of the sub-collection, and the number of words in the vocabulary.

The MEDLINE collection includes 106 queries. Each query is accompanied by a set of relevance judgments selected from the entire collection of documents. The division of the collection of documents into sub-collections leads inevitably to a decrease in the number of relevant documents for each query. In other words, if we have n documents relevant to a given query q with respect to the entire collection, then we will certainly have m documents relevant to the same query with respect to one of the sub-collections, where the value of n is greater or equal to the value of m. Furthermore, the probability of the non-existence of any document relevant to a given query is possible. In this case, each query that does not include any relevant document in a given sub-collection is removed. Table 2 presents, for each sub-collection, the number of queries (Nb Queries), the average query length in terms of number of words (Avr Query Len), and the average number of relevant documents (Avr Rel Doc).

All non-informative words, such as prepositions, conjunctions, pronouns, and very common verbs, are disregarded during the indexing phase. Moreover, the most common morphological and inflectional suffixes are removed using a standard stemming algorithm. In addition, the weights of the words are calculated using the well-known Okapi BM25 term weighting function, presented in Section 2.

Table 1 Summary of sub-collections used in our experiments

Size of the	No. of docs	50,000	100,000	150,000	200,000	250,000	300,000
collection	Mb	26.39	52.36	80.72	107.58	135.05	164.31
Size of dictionary		81, 937	120, 825	156, 009	184, 514	211, 504	237, 889

Table 2 Statistics on the MEDLINE sub-collections queries

No. of docs	50,000	100,000	150,000	200,000	250,000	300,000
Nb Queries	82	91	95	97	99	101
Avr Rel Doc	4.23	7	10.94	13.78	15.5	19.24
Avr Query Len	6.79	6.12	5.68	5.74	5.62	5.51

4.2 Software

The proposed approach was implemented in Python. All the experiments were conducted on a Sony-Vaio workstation having an Intel i3-2330M/2.20 GHz processor and 4 GB RAM and running Ubuntu GNU/Linux 12.04.

4.3 Evaluation metrics

The precision (P) and the mean average precision (MAP) measures were used to evaluate the performance of the proposed approach. The precision is the ratio of relevant documents retrieved over the total number of documents retrieved. It is given by

$$P = \frac{\text{Number of relevant documents retrieved}}{\text{Number of documents retrieved}}. \tag{26}$$

The MAP is the mean of the average precision scores over all queries. It is computed as

$$\text{MAP} = \frac{1}{Nbq}\sum_{i=1}^{Nbq}\frac{1}{m_i}\sum_{j=1}^{m_i}P(R_{ij}), \tag{27}$$

where Nbq is the number of queries, m_i is the number of relevant documents for the ith query, and R_{ij} is the set of ranked retrieval results from the top result until the document d_j is achieved.

4.4 Results

Before proceeding to evaluate the performance of the proposed approach, we first fixed the parameter σ of the kernel functions used to compute the internal correlation. For that purpose, as a preliminary experiment, we considered the internal correlation as the overall correlation and systematically tested a set of fixed σ values from 1 to 40 in increments of 5. Table 3 shows the precision values after retrieving five documents ($P@5$) and the MAP achieved using the sub-collection of $50,000$ documents. The number of pseudo-relevant documents R is tuned at 10, 20, and 50, and the number of expansion keywords is set to 10, which is the typical choice according to Carpineto and Romano [11]. Carpineto and Romano also demonstrated that this number can be increased to 30 keywords. Therefore, in the case where we have candidate keywords that have the same score, we do not attempt to distinguish them and simply add all to the original search query.

As can be seen in Table 3, the suitable values for σ that yield the highest performance are 5 (10 out of 18), 10 (9 out of 18), 30 (7 out of 18), and 25 (5 out of 18). In terms of MAP, the best results are achieved for $\sigma = 30$ in four cases, $\sigma = 10$ in three cases, $\sigma = 5$ in one case, and $\sigma = 25$ in one case.

Table 3 Best performance of the proposed approach for different σ

Kernel	R	σ	1	5	10	15	20	25	30	35	40
Gaussian	10	$P@5$	0.1560	**0.1682**	**0.1682**	0.1658	0.1658	0.1658	0.1658	0.1658	0.1658
		MAP	0.2207	0.2253	**0.2265**	0.2231	0.2230	0.2230	0.2230	0.2230	0.2230
	20	$P@5$	0.1609	**0.1682**	**0.1682**	**0.1682**	**0.1682**	**0.1682**	**0.1682**	**0.1682**	**0.1682**
		MAP	0.2208	0.2252	**0.2255**	0.2245	0.2245	0.2245	0.2245	0.2245	0.2245
	50	$P@5$	0.1609	**0.1682**	0.1658	0.1658	**0.1682**	**0.1682**	**0.1682**	**0.1682**	**0.1682**
		MAP	0.2193	**0.2241**	0.2231	0.2228	0.2235	0.2233	0.2233	0.2233	0.2233
Triangle	10	$P@5$	0.1609	**0.1707**	0.1682	0.1634	0.1682	0.1682	0.1682	0.1682	0.1682
		MAP	0.2110	0.2245	0.2234	0.2250	0.2257	0.2258	**0.2273**	0.2271	0.2252
	20	$P@5$	0.1609	**0.1731**	0.1682	0.1658	0.1682	0.1682	0.1682	0.1658	0.1658
		MAP	0.2110	0.2235	0.2211	0.2234	0.2249	0.2265	**0.2274**	0.2271	0.2252
	50	$P@5$	0.1609	**0.1682**	**0.1682**	0.1658	**0.1682**	**0.1682**	0.1658	0.1634	0.1634
		MAP	0.2110	0.2200	0.2200	0.2235	0.2248	0.2252	**0.2259**	0.2255	0.2235
Cosine	10	$P@5$	0.1609	**0.1682**	**0.1682**	0.1658	**0.1682**	**0.1682**	**0.1682**	**0.1682**	0.1658
		MAP	0.2110	0.2248	0.2255	0.2249	0.2264	0.2261	**0.2278**	0.2255	0.2232
	20	$P@5$	0.1609	**0.1707**	**0.1707**	0.1682	0.1658	0.1658	0.1658	0.1682	0.1682
		MAP	0.2110	0.2239	0.2229	0.2251	0.2255	**0.2267**	0.2251	0.2255	0.2247
	50	$P@5$	0.1609	**0.1682**	**0.1682**	0.1658	0.1658	0.1658	0.1658	0.1658	0.1658
		MAP	0.2110	0.2253	**0.2265**	0.2231	0.2230	0.2230	0.2230	0.2230	0.2230

Table 4_a Comparison of the performance of the EXT/INT, EXT, and INT methods in terms of precision: precision after retrieving 5 documents ($P@5$)

#docs	Performance	EXT/INT			EXT	INT		
		Gaussian	Triangle	Cosine		Gaussian	Triangle	Cosine
100,000	$P@5$	0.1979	0.1845	0.1846	0.1626	0.1758	0.1736	0.1692
	Rate	Gaussian			**+21.65%**	+12.51%	**+13.94%**	**+16.90%**
			Triangle		+13.47%	+4.95%	+6.28%	+9.04%
				Cosine	+13.53%	+5.01%	+6.34%	+9.10%
200,000	$P@5$	0.2432	0.2453	0.2432	0.2164	0.2164	0.2226	0.2247
	Rate	Gaussian			+12.38%	+12.38%	+9.25%	+8.23%
			Triangle		+13.35%	**+13.35%**	+10.20%	+9.17%
				Cosine	+12.38%	+12.38%	+9.25%	+8.23%
300,000	$P@5$	0.2435	0.2475	0.2475	0.2297	0.2415	0.2376	0.2376
	Rate	Gaussian			+6.01%	+0.83%	+2.48%	+2.48%
			Triangle		+7.75%	+2.48%	+4.17%	+4.17%
				Cosine	+7.75%	+2.48%	+4.17%	+4.17%

Table 4_b Comparison of the performance of the EXT/INT, EXT, and INT methods in terms of precision: precision after retrieving 10 documents ($P@10$)

#docs	Performance	EXT/INT			EXT	INT		
		Gaussian	Triangle	Cosine		Gaussian	Triangle	Cosine
100,000	$P@10$	0.1417	0.1406	0.1406	0.1296	0.1351	0.1351	0.1351
	Rate	Gaussian			**+9.34%**	+4.89%	+4.89%	+4.89%
			Triangle		+8.49%	+4.07%	+4.07%	+4.07%
				Cosine	+8.49%	+4.07%	+4.07%	+4.07%
200,000	$P@10$	0.1979	0.1979	0.1979	0.1835	0.1824	0.1824	0.1835
	Rate	Gaussian			+7.85%	**+8.50%**	+7.41%	**+7.85%**
			Triangle		+7.85%	**+8.50%**	**+8.50%**	**+7.85%**
				Cosine	+7.85%	**+8.50%**	**+8.50%**	**+7.85%**
300,000	$P@10$	0.2099	0.2089	0.2079	0.1970	0.2009	0.2059	0.2069
	Rate	Gaussian			+6.55%	+4.48%	+1.94%	+1.45%
			Triangle		+6.04%	+3.98%	+1.46%	+0.97%
				Cosine	+5.53%	+3.38%	+0.97%	+0.48%

In the first phase of comparison, we evaluated the effectiveness of our proposed method through the use of only the external correlation, only the internal correlation, and both the external and internal correlations. In this experiment, the parameter σ was fixed to 5 and both the pseudo-relevant documents and the expansion keywords were set to 10. Tables 4_a and 4_b present, for each sub-collection, the precision values obtained by the external correlation (EXT), the internal correlation (INT), and both the external and internal correlations (EXT/INT) after retrieving five and ten documents. In Table 4, the designation *Rate* indicates the percentage of precision improvement of EXT/INT over EXT and INT.

In Table 4_a, it can clearly be seen that EXT/INT produces the highest $P@5$ values for all sub-collections and achieves a highly significant improvement over EXT and INT (Gaussian, Triangle, Cosine); e.g., on the $200,000$ sub-collection, there is an improvement (by EXT/INT (Triangle)) of 13.35% over EXT, 13.35% over INT (Gaussian), 10.20% over INT (Triangle), and 9.17% over INT (Cosine). Similarly, the relevance precision at ten retrieved documents improves from 0.1835 (+7.85%), 0.1824 (+8.50%), 0.1824 (+8.50%), and 0.1835 (+7.85%) to 0.1979 over EXT, INT (Gaussian), INT (Triangle) and INT (Cosine), respectively. In terms of MAP, we notice that the proposed approach, EXT/INT, shows the best results in all the sub-collections (see Table 5); e.g., on the $300,000$ sub-collection, EXT/INT using Gaussian function outperforms EXT, INT (Gaussian), INT (Triangle), and INT (Cosine), by approximately 3%, 4%, 5%, and 5%, respectively.

In the second set of experiments, we evaluated and compared the results of the proposed approach (EXT/INT), which uses both the external and internal correlations, with those obtained by RSJ, Rocchio, RSV, KLD, and IRF, where we compute the precision values after retrieving five and ten documents. In this experiment, the parameters σ, R, and the number of expansion keywords were set to 5, 10, and 10, respectively. Figure 1 shows the precision values for the EXT/INT, RSJ, Rocchio, RSV, KLD, and IRF techniques.

Table 5 Comparison of the effectiveness of the EXT/INT, EXT, and INT methods in terms of mean average precision

#docs	Performance	EXT/INT			EXT	INT		
		Gaussian	Triangle	Cosine		Gaussian	Triangle	Cosine
100,000	MAP	0.1823	0.1781	0.1791	0.1658	0.1686	0.1719	0.1713
	Rate	Gaussian			**+9.95%**	+8.13%	+6.05%	+6.42%
			Triangle		+7.42%	+5.63%	+3.61%	+3.97%
				Cosine	+8.02%	+6.23%	+4.19%	+4.55%
200,000	MAP	0.1663	0.1684	0.1686	0.1549	0.1531	0.1535	0.1537
	Rate	Gaussian			+7.36%	+8.62%	+8.34%	+8.20%
			Triangle		+8.72%	+9.99%	+3.90%	+9.56%
				Cosine	+8.84%	**+10.12%**	**+9.84%**	**+9.69%**
300,000	MAP	0.1617	0.1607	0.1607	0.1556	0.1554	0.1526	0.1527
	Rate	Gaussian			+3.92%	+4.05%	+5.96%	+5.89%
			Triangle		+3.28%	+3.41%	+5.31%	+5.24%
				Cosine	+3.28%	+3.41%	+5.31%	+5.24%

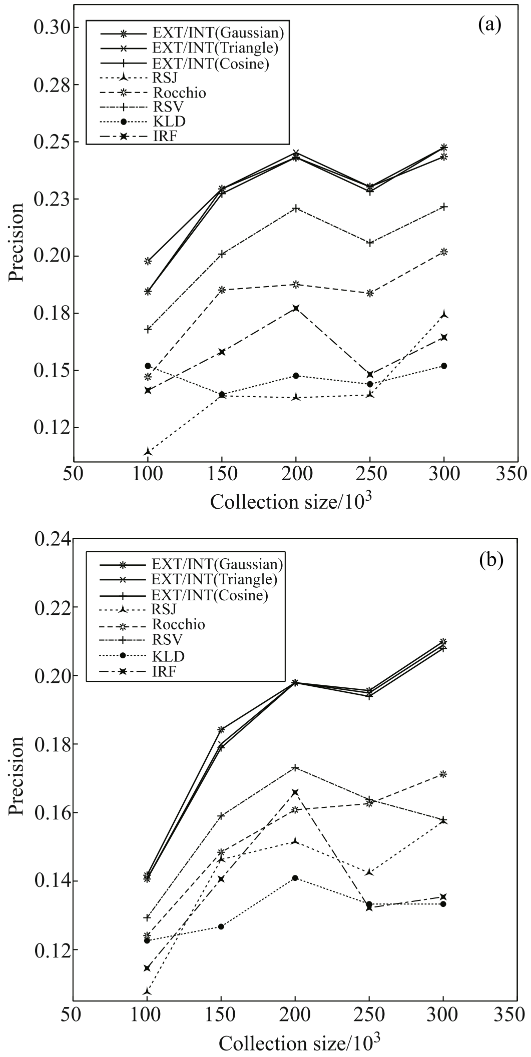

Fig. 1 Effectiveness comparison of the EXT/INT approach and the RSJ, Rocchio, RSV, KLD, and IRF methods in terms of precision. (a) Precision after retrieving five documents ($P@5$); (b) precision after retrieving ten documents ($P@10$)

In Fig. 1(a), we can see a clear superiority of the proposed approach, EXT/INT, over Rocchio and RSV, and this superiority is more significant in comparison with the RSJ, KLD, and IRF techniques. It is clearly seen in Fig. 1(a) that the proposed approach succeeds in improving the search results, after retrieving five documents, in all the sub-collections; e.g., on the 300, 000 sub-collection, EXT/INT using Cosine shows a great improvement of +42.08% over RSJ, +22.59% over Rocchio, +11.69% over RSV, +62.83% over KLD, and +50.45% over IRF. Despite the superiority shown in Fig. 1(b), the results are not similar to those observed in Fig. 1(a). Nevertheless, the precision values of the proposed approach after retrieving ten documents are the best in all the sub-collections.

From the values in Table 4 and Fig. 1, we can conclude that the proposed method, EXT/INT, succeeds in improving the ranking of the relevant documents and puts them in the first place. The precision values of the proposed system, after retrieving five documents, show a clear and significant superiority of our method to the EXT, INT, RSJ, Rocchio, RSV, KLD, and IRF techniques. This confirms the effectiveness of the EXT/INT approach.

In the next phase of testing, we computed the MAP to evaluate the retrieval effectiveness of the EXT/INT and the PRF methods (Table 6). The parameters σ, R, and the number of expansion keywords were set to 5, 10, and 10, respectively. Furthermore, we used the two-tailed t-test to measure the statistical significance of the differences between the MAP values.

Table 6 shows a clear advantage of the EXT/INT approach as compared to the RSJ, Rocchio, RSV, and KLD approaches. The improvements over RSJ, Rocchio, RSV, KLD, and IRF are statistically significant in 64 out of 75 cases ($p < 0.05$), and 75 out of 75 improvements are positive;

e.g., on the 300000 sub-collection, EXT/INT (Gaussian) outperforms RSJ by +20.31%, Rocchio by +21.31%, RSV by +28.12%, KLD by +94.11%, and IRF by +54.76%, while EXT/INT (Triangle) and EXT/INT (Cosine) outperform RSJ by +19.57%, Rocchio by +20.56%, RSV by +27.33%, KLD by +92.91%, and IRF by +53.78%, respectively. The values in bold represent the best improvement achieved by the proposed approach.

In the final set of experiments, we compared the performance of EXT/INT with that of the RSJ, Rocchio, RSV, KLD, and IRF techniques while varying the values of σ and R and the number of expansion keywords. As a first step, we changed the value of σ from 5 to 30 and maintained the parameter R and the number of expansion keywords constant at 10 (Table 7). In the second step, we set σ and the number of expansion keywords to 5 and 10, and varied the value of R between 20 and 50 (Tables 8_a and 8_b). In the final step, we compared EXT/INT with RSJ, Rocchio, RSV, KLD, and IRF while varying the number of expansion terms between 10 and 5 and tuned the parameters σ and R to 5 and 10, respectively (Table 9).

Once again, the proposed approach EXT/INT achieved a better performance than the other methods, although the number of pseudo-relevant documents R, the parameter σ, and the number of expansion terms were varied. In terms of precision, the EXT/INT results are the best in all the sub-collections, and in terms of MAP, EXT/INT performed statistically significantly better than RSJ, Rocchio, RSV,

Table 6 Mean Average Precision (MAP) results of EXT/INT, RSJ, Rocchio, RSV, KLD, and IRF methods

#docs	Performance	EXT/INT			RSJ	Rocchio	RSV	KLD	IRF
		Gaussian	Triangle	Cosine					
100,000	MAP	0.1823	0.1781	0.1791	0.1253	0.1524	0.1653	0.1236	0.1226
		Gaussian			**+45.49%**∗	+19.62%∗	+10.28%	+47.49%∗	+48.69%∗
	Rate		Triangle		+42.14%∗	+16.86%	+07.18%	+44.09%∗	+45.27%∗
				Cosine	+42.94%∗	+17.52%	+08.34%	+44.90%∗	+46.08%∗
150,000	MAP	0.1763	0.1784	0.1781	0.1285	0.1540	0.1460	0.0990	0.1130
		Gaussian			+37.20%∗	+14.48%	+20.75%∗	+78.08%∗	+56.01%∗
	Rate		Triangle		+38.83%∗	+15.84%	+22.19%∗	+80.20%∗	+57.88%∗
				Cosine	+38.60%∗	+15.65%	+21.98%∗	+79.89%∗	+57.61%∗
200,000	MAP	0.1663	0.1684	0.1686	0.1174	0.1346	0.1326	0.0924	0.1200
		Gaussian			+41.65%∗	+23.55%∗	+25.41%∗	+79.97%∗	+38.58%∗
	Rate		Triangle		+43.44%∗	+25.11%∗	+26.99%∗	+82.24%∗	+40.34%∗
				Cosine	+43.61%∗	**+25.26%**∗	+27.14%∗	+82.46%∗	+40.50%∗
250,000	MAP	0.1599	0.1585	0.1585	0.1204	0.1302	0.1378	0.0796	0.0886
		Gaussian			+32.81%∗	+22.81%∗	+16.03%	**+100.87%***	**+80.47%**∗
	Rate		Triangle		+31.64%∗	+21.74%∗	+15.02%	+99.21%∗	+78.89%∗
				Cosine	+31.64%∗	+21.74%∗	+15.02%	+99.21%∗	+78.89%∗
300,000	MAP	0.1617	0.1607	0.1607	0.1344	0.1333	0.1262	0.0833	0.1045
		Gaussian			+20.31%∗	+21.31%∗	**+28.12%***	+94.11%∗	+54.76%∗
	Rate		Triangle		+19.57%∗	+20.56%∗	+27.33%∗	+92.91%∗	+53.78%∗
				Cosine	+19.57%∗	+20.56%∗	+27.33%∗	+92.91%∗	+53.78%∗

Note: ∗ indicates the difference is statistically significant, p-value < 0.05 with two-tailed t-test

Table 7 Comparison of the performance of EXT/INT, RSJ, Rocchio, RSV, KLD, and IRF methods (σ = 30)

#docs	Performance	EXT/INT			RSJ	Rocchio	RSV	KLD	IRF
		Gaussian	Triangle	Cosine					
100,000	P@5	**0.1978**	0.1956	0.1956	0.1142	0.1472	0.1880	0.1520	0.1413
	P@10	**0.1428**	**0.1428**	**0.1428**	0.1076	0.1241	0.1293	0.1226	0.1146
	MAP	**0.1817**	0.1809	**0.1817**	0.1253∗	0.1524	0.1653	0.1236∗	0.1226*
200,000	P@5	**0.2453**	0.2432	**0.2453**	0.1381	0.1876	0.2209	0.1477	0.1772
	P@10	**0.1979**	**0.1979**	**0.1979**	0.1515	0.1608	0.1731	0.1409	0.1659
	MAP	0.1654	**0.1662**	0.1658	0.1174∗	0.1346∗	0.1326∗	0.0924∗	0.1200*
300,000	P@5	0.2415	0.2435	**0.2455**	0.1742	0.2019	0.2216	0.1520	0.1645
	P@10	0.2069	**0.2079**	0.2069	0.1574	0.1712	0.1579	0.1333	0.1354
	MAP	**0.1617**	0.1609	0.1614	0.1344∗	0.1333∗	0.1262∗	0.0833∗	0.1045*

Note: ∗ indicates the difference is statistically significant, p-value < 0.05 with two-tailed t-test

Table 8_a Precision and mean average precision results of EXT/INT, RSJ, Rocchio, RSV, KLD, and IRF approaches: effectiveness comparison of EXT/INT with the baseline, $R = 20$

#docs	Performance	EXT/INT			RSJ	Rocchio	RSV	KLD	IRF
		Gaussian	Triangle	Cosine					
100,000	*P*@5	**0.1978**	0.1846	0.1846	0.1230	0.1582	0.1826	0.1200	0.1573
	MAP	**0.1813**	0.1785	0.1790	0.1358∗	0.1512	0.1620	0.1160∗	0.1379*
200,000	*P*@5	**0.2453**	0.2412	0.2412	0.1546	0.2000	0.2163	0.1409	0.1613
	MAP	0.1641	0.1662	**0.1680**	0.1405	0.1339∗	0.1308∗	0.0870∗	0.1108*
300,000	*P*@5	0.2415	**0.2455**	**0.2455**	0.1742	0.2198	0.2216	0.1375	0.1520
	MAP	0.1633	**0.1655**	0.1645	0.1496	0.1291∗	0.1261∗	0.0859∗	0.0845*

Note: ∗ indicates the difference is statistically significant, p-value < 0.05 with two-tailed t-test

Table 8_b Precision and mean average precision results of EXT/INT, RSJ, Rocchio, RSV, KLD, and IRF approaches: effectiveness comparison of EXT/INT with the baseline, $R = 50$

#docs	Performance	EXT/INT			RSJ	Rocchio	RSV	KLD	IRF
		Gaussian	Triangle	Cosine					
100,000	*P*@5	**0.1890**	0.1824	0.1824	0.1406	0.1274	0.1746	0.1200	0.1333
	MAP	**0.1786**	0.1742	0.1747	0.1427∗	0.1394∗	0.1612	0.1119∗	0.1099*
200,000	*P*@5	**0.2329**	0.2309	0.2288	0.1711	0.1649	0.2186	0.1340	0.1340
	MAP	0.1597	**0.1633**	0.1628	0.1383∗	0.1169∗	0.1312∗	0.0869∗	0.0907*
300,000	*P*@5	0.2396	**0.2435**	**0.2435**	0.1762	0.1920	0.2133	0.1291	0.1458
	MAP	0.1608	0.1618	**0.1619**	0.1447	0.1121∗	0.1232∗	0.0831∗	0.0853*

Note: ∗ indicates the difference is statistically significant, p-value < 0.05 with two-tailed t-test

Table 9 Comparison of the performance of the EXT/INT, RSJ, Rocchio, RSV, KLD, and IRF methods

#docs	Performance	EXT/INT			RSJ	Rocchio	RSV	KLD	IRF
		Gaussian	Triangle	Cosine					
100000	*P*@5	**0.1736**	0.1692	0.1670	0.0857	0.1670	0.1880	0.1653	0.1413
	P@10	**0.1384**	0.1362	0.1362	0.1087	0.1252	0.1320	0.1280	0.1293
	MAP	0.1717	0.1726	**0.1727**	0.1150∗	0.1626	0.1674	0.1444	0.1401*
200000	*P*@5	0.2185	**0.2247**	**0.2247**	0.1195	0.2000	0.2231	0.1750	0.2022
	P@10	**0.1845**	0.1835	0.1835	0.1525	0.1639	0.1754	0.1613	0.1659
	MAP	**0.1543**	0.1529	0.1517	0.0993∗	0.1465	0.1349	0.1078∗	0.1353
300000	*P*@5	**0.2435**	0.2376	0.2356	0.1306	0.2099	0.2237	0.1770	0.1937
	P@10	0.2029	**0.2079**	**0.2079**	0.1623	0.1772	0.1831	0.1593	0.1614
	MAP	0.1540	**0.1559**	**0.1559**	0.1036∗	0.1338∗	0.1488	0.1036∗	0.1181*

Note: ∗ indicates the difference is statistically significant, p-value < 0.05 with two-tailed t-test

KLD, and IRF in a considerable number of cases.

5 Conclusion

In this paper, we proposed two criteria to assess the degree of relatedness between a candidate expansion term and the query keywords: the co-occurrence and the proximity. We adopted the strength Pareto fitness assignment to satisfy both criteria simultaneously. We also introduced the concept of the external/internal correlation of terms. This concept, which involves the GTD probability and the well-known kernel functions, is based on finding for each query term the locations where it occurs and then selecting from these locations the words that frequently neighbor and co-occur with that query term. The original query is then expanded by adding the top selected terms and re-interrogated using the Okapi BM25 document-scoring function.

We tested our approach in depth using the MEDLINE dataset. The experimental results show that the proposed approach, EXT/INT, succeeds in improving the ranking of the relevant documents and yields a substantial enhancement in terms of precision and MAP as compared to the baseline.

Future work in this area will include exploiting the semantic aspect of keywords in order to further enhance the retrieval effectiveness. It will also be interesting to test the proposed approach on other existing medical datasets, such as TREC CDS and CLEF eHealth. Another possible research direction is to extend our work to additional areas of search applications, such as Twitter search.

References

1. Ranganathan P. From microprocessors to nanostores: rethinking data-centric systems. IEEE Computer, 2011, 44(1): 39–48
2. Zhu Y Y, Zhong N, Xiong Y. Data explosion, data nature and dataology. In: Proceedings of International Conference on Brain Informatics. 2009, 147–158
3. Ntoulas A, Cho J, Olston C. What's new on the Web?: the evolution of the Web from a search engine perspective. In: Proceedings of the 13th International Conference on World Wide Web. 2004, 1–12
4. Bharat K, Broder A. A technique for measuring the relative size and overlap of public web search engines. Computer Networks and ISDN Systems, 1998, 30(1): 379–388
5. Williams H E, Zobel J. Searchable words on the Web. International Journal on Digital Libraries, 2005, 5(2): 99–105
6. Eisenstein J, O'Connor B, Smith N A, Xing E P. Mapping the geographical diffusion of new words. In: Proceedings of Workshop on Social Network and Social Media Analysis: Methods, Models and Applications. 2012
7. Sun H M. A study of the features of internet english from the linguistic perspective. Studies in Literature and Language, 2010, 1(7): 98–103
8. Chen Q, Li M, Zhou M. Improving query spelling correction using Web search results. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2007, 181–189
9. Subramaniam L V, Roy S, Faruquie T A, Negi S. A survey of types of text noise and techniques to handle noisy text. In: Proceedings of the 3rd Workshop on Analytics for Noisy Unstructured Text Data. 2009, 115–122
10. Ahmad F, Kondrak G. Learning a spelling error model from search query logs. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. 2005, 955–962
11. Carpineto C, Romano G. A survey of automatic query expansion in information retrieval. ACM Computing Surveys, 2012, 44(1): 1–50
12. Véronis J. Hyperlex: lexical cartography for information retrieval. Computer Speech & Language, 2004, 18(3): 223–252
13. Bernardini A, Carpineto C, Amico M D. Full-subtopic retrieval with keyphrase-based search results clustering. In: Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technologies. 2009, 206–213
14. Wong S K M, Ziarko W, Raghavan V V, Wong P. On modeling of information retrieval concepts in vector spaces. ACM Transactions on Database Systems, 1987, 12(2): 299–321
15. Crestani F. Application of spreading activation techniques in information retrieval. Artificial Intelligence Review, 1997, 11(6): 453–482
16. Carpineto C, Romano G. Concept Data Analysis: Theory and Applications. Chichester: John Wiley & Sons, 2004
17. Sahlgren M. An introduction to random indexing. In: Proceedings of Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering. 2005
18. Melucci M. A basis for information retrieval in context. ACM Transactions on Information Systems, 2008, 26(3): 1–41
19. Sun R, Ong C H, Chua T S. Mining dependency relations for query expansion in passage retrieval. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2006, 382–389
20. Schlaefer N, Ko J, Betteridge J, Pathak M A, Nyberg E, Sautter G. Semantic extensions of the Ephyra QA system for TREC 2007. In: Proceedings of the 16th Text REtrieval Conference. 2007
21. Kraaij W, Nie J Y, Simard M. Embedding Web-based statistical translation models in cross-language information retrieval. Computational Linguistics, 2003, 29(3): 381–419
22. Kherfi M L, Ziou D, Bernardi A. Image retrieval from the World Wide Web: issues, techniques, and systems. ACM Computing Surveys, 2004, 36(1): 35–67
23. Natsev A P, Haubold A, Tešić J, Xie L X, Yan R. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In: Proceedings of the 15th ACM International Conference on Multimedia. 2007, 991–1000
24. Arguello J, Elsas J L, Callan J, Carbonell J G. Document representation and query expansion models for blog recommendation. In: Proceedings of the 2nd International Conference on Weblogs and Social Media. 2008, 10–18
25. Hidalgo J M G, de Buenaga Rodríguez M, Pérez J C C. The role of word sense disambiguation in automated text categorization. In: Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems. 2005, 298–309
26. Graupmann J, Cai J, Schenkel R. Automatic query refinement using mined semantic relations. In: Proceedings of International Workshop on Challenges in Web Information Retrieval and Integration. 2005, 205–213
27. Kamvar M, Baluja S. The role of context in query input: using contextual signals to complete queries on mobile devices. In: Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services. 2007, 405–412
28. Huang C C, Lin K M, Chien L F. Automatic training corpora acquisition through Web mining. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technologies. 2005, 193–199
29. Perugini S, Ramakrishnan N. Interacting with Web hierarchies. IT Professional, 2006, 8(4): 19–28
30. Church K, Smyth B. Mobile content enrichment. In: Proceedings of the 12th International Conference on Intelligent User Interfaces. 2007, 112–121
31. Macdonald C, Ounis I. Expertise drift and query expansion in expert search. In: Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management. 2007, 341–350
32. Billerbeck B, Zobel J. Document expansion versus query expansion for ad-hoc retrieval. In: Proceedings of the 10th Australasian Document Computing Symposium. 2005, 34–41
33. Shokouhi M, Azzopardi L, Thomas P. Effective query expansion for federated search. In: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2009, 427–434
34. Wang H, Liang Y, Fu L, Xue G R, Yu Y. Efficient query expansion

for advertisement search. In: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2009, 51–58

35. Voorhees E M. Query expansion using lexical-semantic relations. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1994, 61–69

36. Collins-Thompson K, Callan J. Query expansion using random walk models. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management. 2005, 704–711

37. Liu S, Liu F, Yu C, Meng W Y. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2004, 266–272

38. Song M, Song I Y, Hu X H, Allen R B. Integration of association rules and ontologies for semantic query expansion. Data & Knowledge Engineering, 2007, 63(1): 63–75

39. Gauch S, Wang J Y, Rachakonda S M. A corpus analysis approach for automatic query expansion and its extension to multiple databases. ACM Transactions on Information Systems, 1999, 17(3): 250–269

40. Hu J N, Deng W H, Guo J. Improving retrieval performance by global analysis. In: Proceedings of the 18th International Conference on Pattern Recognition. 2006, 703–706

41. Park L A, Ramamohanarao K. Query expansion using a collection dependent probabilistic latent semantic thesaurus. In: Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. 2007, 224–235

42. Milne D N, Witten I H, Nichols D M. A knowledge-based search engine powered by wikipedia. In: Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management. 2007, 445–454

43. Rocchio J J. Relevance feedback in information retrieval. The SMART Retrieval System-Experiments in Automatic Document Processing, 1971, 313–323

44. Robertson S E, Jones K S. Relevance weighting of search terms. Journal of the American Society for Information Science, 1976, 27(3): 129–146

45. Wong W, Luk R W P, Leong H V, Ho K, Lee D L. Re-examining the effects of adding relevance information in a relevance feedback environment. Information Processing & Management, 2008, 44(3): 1086–1116

46. Zhai C X, Lafferty J. Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of the 10th International Conference on Information and Knowledge Management. 2001, 403–410

47. Lavrenko V, Croft W B. Relevance based language models. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2001, 120–127

48. Khennak I, Drias H. Strength pareto fitness assignment for generating expansion features. In: Proceedings of the 3rd World Conference on Information Systems and Technologies. 2015, 133–142

49. Robertson S, Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends® in Information Retrieval, 2009, 3(4): 333–389

50. Robertson S E. On term selection for query expansion. Journal of Documentation, 1990, 46(4): 359–364

51. Carpineto C, De Mori R, Romano G, Bigi B. An information-theoretic approach to automatic query expansion. ACM Transactions on Information Systems, 2001, 19(1): 1–27

52. Jurafsky D, Martin J H. Speech and Language Processing. Upper Saddle River, NJ: Pearson Prentice Hall, 2014

Ilyes Khennak is a PhD student in computer science at University of Sciences and Technology Houari Boumediene (USTHB), Algeria. He received his master degree in intelligent computer systems from USTHB in 2011. His research interests include artificial intelligence and information retrieval.

Habiba Drias received the MS degree in computer science from Case Western Reserve University, USA in 1984 and the PhD degree in computer science from University of Sciences and Technology Houari Boumediene (USTHB), Algeria in collaboration with UPMC, France in 1993. She is currently a full professor at USTHB since 1999 and directs the Laboratory of Research in Artificial Intelligence (LRIA). She has published around 200 papers in well-recognized international conference proceedings and journals and has directed 20 PhD theses, 38 master theses and 31 engineer projects. In 2013, she won the Algerian Scopus award in computer science, and she was selected by a jury of international academicians as a founding member of the Algerian Academy of Science and Technology (AAST) in 2015.