Improving Relevance Feedback-Based Query Expansion by the Use of a Weighted Word Pairs Approach

Francesco Colace
DIEM, Università degli Studi di Salerno, Via Papa Giovanni Paolo II 132, Fisciano, Salerno 84084, Italy. E-mail: fcolace@unisa.it

Massimo De Santo
DIEM, Università degli Studi di Salerno, Via Papa Giovanni Paolo II 132, Fisciano, Salerno 84084, Italy. E-mail: desanto@unisa.it

Luca Greco
DIEM, Università degli Studi di Salerno, Via Papa Giovanni Paolo II 132, Fisciano, Salerno 84084, Italy. E-mail: lgreco@unisa.it

Paolo Napoletano
Department of Informatics, Systems and Communication, Università Degli Studi di Milano Bicocca, Milano 20126, Italy. E-mail: napoletano@disco.unimib.it

In this article, the use of a new term extraction method for query expansion (QE) in text retrieval is investigated. The new method expands the initial query with a structured representation made of weighted word pairs (WWP) extracted from a set of training documents (relevance feedback). Standard text retrieval systems can handle a WWP structure through custom Boolean weighted models. We experimented with both the explicit and pseudorelevance feedback schemas and compared the proposed term extraction method with others in the literature, such as KLD and RM3. Evaluations have been conducted on a number of test collections (Text REtrivel Conference [TREC]-6, -7, -8, -9, and -10). Results demonstrated that the QE method based on this new structure outperforms the baseline.

Introduction

In the field of text retrieval, a typical problem is: "How can a system tell which documents are relevant to a query? Which results are more relevant than others?" To answer these questions, several information retrieval (IR) models have been proposed: set-theoretic (including Boolean), algebraic, and probabilistic models (Manning, Raghavan, & Schütze, 2008; Baeza-Yates & Ribeiro-Neto, 1999). Although each technique has its own properties, most methods rely on the "bag of words" model for document and query representation.

The bag of words assumption claims that a document, as well as a query, can be considered as a feature vector where each element indicates the presence (or absence) of a word, so that the information on the position of that word within the document is completely lost (Manning et al., 2008); the elements of the vector can be weights computed in different ways. The relevance of a document to a query can be determined as the distance between the corresponding vector representations in the space of features.

It has been observed that queries performed by common users may not be long enough (two or three words, on average) (Jansen, Spink, & Saracevic, 2000; Jansen, Booth, & Spink, 2008) to avoid inherent ambiguity of language (polysemy, and so on). So, text retrieval systems, which rely on a term-frequency-based index, generally suffer from low-precision or low-quality document retrieval. The aim of query expansion in IR systems is to reduce this query/document mismatch by expanding the base query using words or phrases with a similar meaning or some other statistical relation to the set of relevant documents (Carpineto, de Mori, Romano, & Bigi, 2001).

In this work, we propose a new query expansion (QE) method that automatically extracts a set of weighted word pairs (WWP) from a set of topic-related documents provided

Received October 7, 2013; revised April 30, 2014; accepted April 30, 2014

 DOI: 10.1002/asi.23331

by the relevance feedback. Such a structured set of terms is obtained by using a method of term extraction previously investigated (Clarizia, Greco, & Napoletano, 2011; Colace, De Santo, Greco, & Napoletano, 2013a, 2013b, 2014) and based on the latent Dirichlet allocation (LDA) model (Blei, Ng, & Jordan, 2003) implemented as the probabilistic topic model (Griffiths, Steyvers, & Tenenbaum, 2007).

Several existing text retrieval systems can handle a WWP structure through custom Boolean weighted models. We were able to test this QE method on the following text retrieval systems: Lucene[1]; Indri[2] (Ogilvie & Callan, 2002; Strohman, Metzler, Turtle, & Croft, 2005); Terrier (TERabyte RetrIEveR)[3] (Ounis et al., 2006); and Zettair.[4] We experimented with both the explicit and pseudorelevance feedback (PRF) schemas and compared the proposed term extraction method with state-of-the-art QE methods, such as Kullback-Leibler divergence (KLD) (Carpineto et al., 2001) and the interpolated version of relevance model (RM3) (Abdul-Jaleel et al., 2004). Evaluations have been conducted on a number of test collections (Text REtrivel Conference [TREC]-6, -7, -8, -9, and -10). Results demonstrate that the QE method based on this new structure outperforms the baseline.

Background and Related Works

Because we propose an alternative approach to term selection for QE, in this section, we discuss the most common methods employed in QE problems with relevance feedback, paying particular attention to related works on term extraction.

The idea of taking advantage of additional knowledge to retrieve relevant documents has been widely discussed in the literature, where manual, interactive, and automatic techniques have been proposed (Baeza-Yates & Ribeiro-Neto, 1999; Carpineto & Romano, 2012; Chang, Colace, Zhao, & Sun, 2011; Efthimiadis, 1996; Manning et al., 2008; Na, Kang, Roh, & Lee, 2005).

A better specialization of the query can be obtained with additional knowledge, which is typically extracted from exogenous (e.g., WordNet) or endogenous knowledge (i.e., extracted only from the documents contained in the collection) (Bhogal, MacFarlane, & Smith, 2007; Manning et al., 2008).

In this work, we focus mainly on those QE techniques that make use of relevance feedback. We can distinguish between three types of procedures for relevance assignment: explicit feedback; implicit feedback; and pseudofeedback (Baeza-Yates & Ribeiro-Neto, 1999). The feedback is usually obtained from assessors and indicates the relevance degree for a document retrieved in response to a query. If the assessors know that the provided feedback will be used as a relevance judgment, then the feedback is called explicit. Implicit feedback is otherwise inferred from user behavior: It takes into account which documents they do and do not select for viewing, the duration of time spent viewing a document, or page browsing or scrolling actions. PRF (or blind feedback) assumes that the top "n" ranked documents obtained after performing the initial query are relevant: This approach is generally used in automatic systems.

Because the human labeling task is enormously boring and time-consuming (Ko & Seo, 2009), some QE methods, which make use of a set of pseudorelevant documents (PRDs), have been proposed in the past. Early work includes concept-based methods (Qiu & Frei, 1993), Phrasefinder (Jing & Croft, 1994), and local context analysis (Xu & Croft, 1996); these approaches use mainly co-occurrence information extracted from the whole target corpus or from a set of top-ranked documents retrieved in response to the base query.

A more recent co-occurrence-based approach relies on relevance-based language models (Lavrenko & Croft, 2001): The query and the relevant documents are assumed to be generated from an underlying relevance model. The model itself is estimated based on the PRD for a query or, in the refined version, called RM3, by incorporating also the original query (Abdul-Jaleel et al., 2004). RM3 is often referred to as one of the most effective automatic QE methods. Another important technique was introduced previously (Carpineto & Romano, 2012; Carpineto et al., 2001), where an effective QE method based on information theoretical principles is proposed: It relies on the KLD between the probability distributions of terms in the relevant documents and in the complete corpus.

In most cases, it can been seen that the reformulated query consists in a simple (sometimes weighted) list of words. Although such methods have proven their effectiveness in terms of accuracy and computational cost, several more complex alternative methods have been proposed, which consider the extraction of a structured set of words instead of a simple list of them: a weighted set of clauses combined with suitable operators (Callan, Croft, & Harding, 1992; Collins-Thompson & Callan, 2005; Lang, Metzler, Wang, & Li, 2010; Metzler & Croft, 2007). Other proposed methods are based on language modeling to integrate several contextual factors in order to adapt document ranking to the specific query context (Bai & Nie, 2008) or integrate term relationships (Bai, Song, Bruza, Nie, & Cao, 2005). Latent semantic analysis has been extensively used in IR, especially for term correlations computing (Park & Ramamohanarao, 2009). Furthermore, several existing term selection methods use language models combined with exogenous knowledge, such as thesaurus (Cao, Nie, & Bai, 2005), WordNet (Pinto, Farina Martinez, & Perez-Sanjulian, 2008; Zhang, Deng, & Li, 2009), or ontology (Bhogal et al., 2007).

Because fully automatic methods can exhibit low performance when the initial query is intrinsically ambiguous, in recent years, some hybrid techniques have been developed that take into account a minimal explicit human feedback

[1]https://lucene.apache.org/
[2]http://www.lemurproject.org/indri.php
[3]http://terrier.org
[4]http://www.seg.rmit.edu.au/zettair/

DOI: 10.1002/asi

(Okabe & Yamada, 2007; Dumais, Joachims, Bharat, & Weigend, 2003) and use it to automatically identify other topic-related documents. Such methods use many documents as feedback (about 40) and achieve a mean average precision of about 30% (Okabe & Yamada, 2007). We will show that the proposed method achieves the same performance of hybrid techniques, but using the same minimal explicit feedback.

Problem Formulation

According to the IR theory, the vector space model (Manning et al., 2008) is an effective way for representing text contents. In fact, a document **d** (as well as a query **q**) can be represented as a vector of weighted words belonging to a vocabulary T (of size $|T|$):

$$\mathbf{d} = (w_1, \ldots, w_{|T|}).$$

Each weight w_n is such that $0 \leq w_n \leq 1$ and represents how much the term t_n contributes to the document **d** (in the same way for **q**). In the term frequency-inverse document frequency (*tf-idf*) model, the weight is typically proportional to the term frequency and inversely proportional to the frequency and length of the documents containing the term.

Given a query, the IR system assigns the relevance to each document of the collection with respect to the query by using a similarity function, as defined in the following Equation (1):

$$sim(\mathbf{q}, \mathbf{d}) = \sum_{t \in \mathbf{q} \cap \mathbf{d}} w_{t,\mathbf{q}} \cdot w_{t,\mathbf{d}}, \quad (1)$$

where $w_{t,\mathbf{q}}$ and $w_{t,\mathbf{d}}$ are the weights of the term t in the query **q** and document **d**, respectively.

QE by Relevance Feedback

The performance of IR systems can be improved by expanding the initial query with other topics-related terms. These QE terms can be manually typed or extracted from feedback documents selected by the user himself (explicit relevance feedback) or automatically chosen by the system (PRF) (Baeza-Yates & Ribeiro-Neto, 1999).

In our view, a general QE framework is a modular system including the following modules:

- Information Retrieval (IR);
- Feedback (F);
- Feature Extraction (FE); and
- Query Reformulation (QR).

A general scheme is represented in Figure 1 and can be explained as follows. Let's consider a generic IR system and a collection of indexed documents D. The user performs a search in the IR system by typing a query **q**. The IR system computes the relevance of each document of the corpus with respect to the query through Equation (1). As a result of the search, a set of ranked documents $\Omega_{res} = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\} \subseteq D$ is returned to the user.

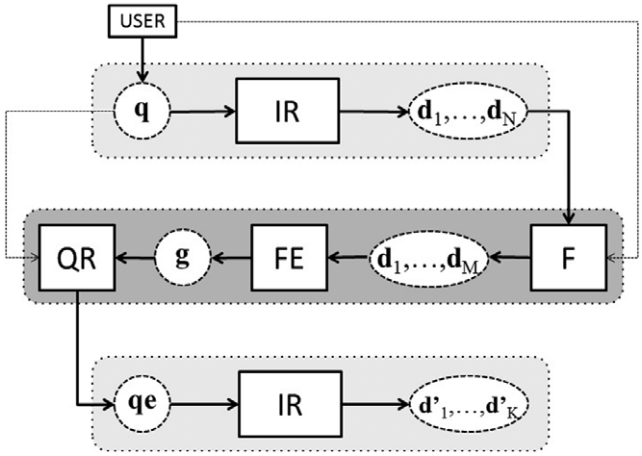

FIG. 1. General framework for QE.

Once the result is available, the module F assigns a judgement of relevance, also known as relevance feedback, to each document of Ω_{res}. The relevance can be manually or automatically (pseudorelevance) assigned. In the case of manual, the user provides the explicit feedback by assigning a positive judgment of relevance to a subset of documents $\Omega_{fback} = \{\mathbf{d}_1, \ldots, \mathbf{d}_M\} \subseteq \Omega_{res}$. In the case of automatic feedback, the module F arbitrarily assigns a positive judgment of relevance to a subset of documents, usually the top M documents retrieved from Ω_{res}.

Given the set of relevant documents Ω_{fback}, the module FE selects a set of features **g** that are then added to the initial query **q**. The selected features can be weighted words or more complex structures, such as the WWP proposed in this article. The QR module adapts the resulting set of features **g** so that they can be added to the initial query and then handled by the IR system. The new expanded query **qe** is then given as input to the IR system in order to perform a new search. As a result, a new set of documents $\Omega_{res} = \{\mathbf{d}'_1, \cdots, \mathbf{d}'_K\}$ is retrieved.

The QE framework just described is quite general. We could use any of the existing IR systems, as well as any of the existing methods, for feature extraction. According to this framework, it is possible to make objective comparisons between different system configurations. In this article, we propose a new FE method for QE (WWP) that has been compared with established state-of-art QE approaches: KLD (Carpineto et al., 2001) and RM3 (Abdul-Jaleel et al., 2004).

We tested our method by considering different open-source IR systems:

- Lucene, a text search engine library part of the Apache Software Foundation. We used the applications bundled with the library to index the considered collections.
- Indri, a search engine built on top of the Lemur (Ogilvie & Callan, 2002) project. This toolkit was designed for research

DOI: 10.1002/asi

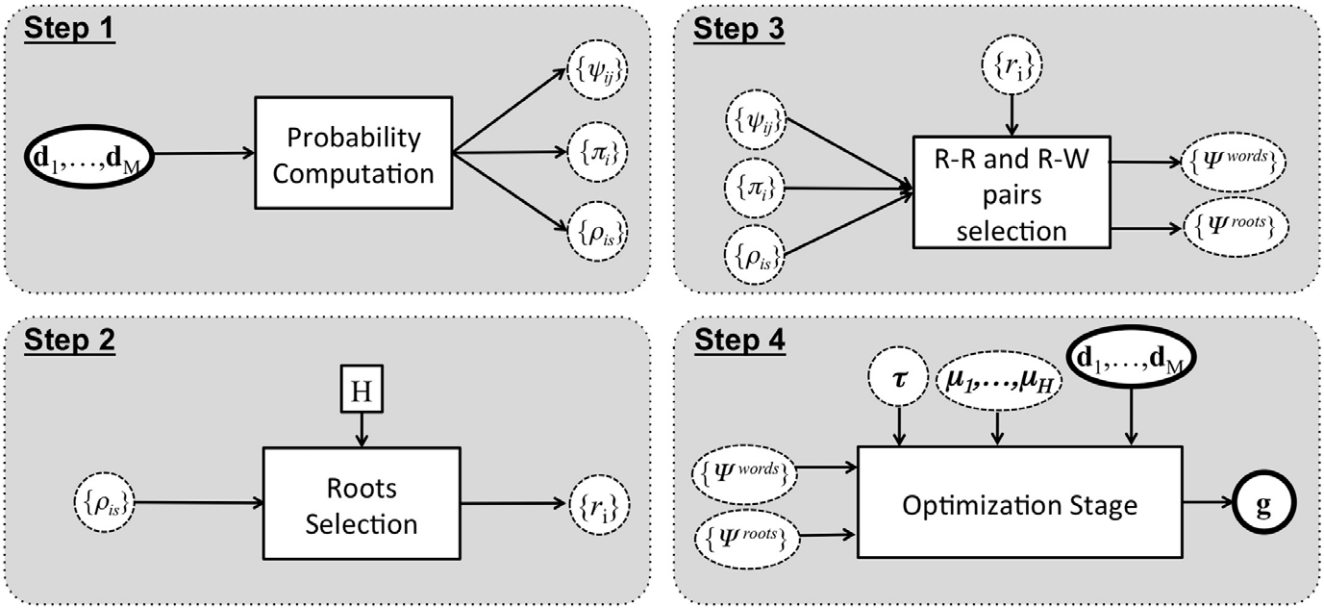

FIG. 2. Steps of the proposed FE method.

in language modeling and IR. This project was developed collaboratively by the University of Massachusetts and Carnegie Mellon University.
- Terrier (TERabyte RetrIEveR), a modular platform that allows rapid development of web, intranet, and desktop search engines, developed at the University of Glasgow. It allows to index, query, and evaluate standard TREC collections.
- Zettair, a text search engine developed by the Search Engine Group at RMIT University. It allows to handle large amounts of text.

The Proposed WWP Extraction Method

The input of the feature extraction module is the set of documents Ω_{fback} and the output is the vector

$$\mathbf{g} = (b_1, \ldots, b_{|G|})$$

containing the weights of all possible $|G|$ word pairs $\{(v, u)_{p|p=1\ldots|G|}\}$. The entire extraction process is divided into four steps and is shown in Figure 2.

Step 1: Probabilities Computation

The input of this step is the set of documents $\Omega_{fback} = \{\mathbf{d}_1, \ldots, \mathbf{d}_M\}$, where each document is represented as a vector of weights. Each weight is associated to a word of the vocabulary T. The outputs of this step are:

1. The a priori probability that a word v_i occurs in Ω_{fback}: $\pi_i = P(v_i)$, $\forall v_i \in T$;
2. The conditional probability that a word v_i occurs in Ω_{fback} given that another word v_s occurred in Ω_{fback}: $\rho_{is} = P(v_i|v_s)$, $\forall v_i, v_s \in T$ and $v_i \neq v_s$; and
3. The joint probability that a pair of words, v_i and v_j, occurs at the same time in Ω_{fback}: $\psi_{ij} = P(v_i, v_j)$, $\forall v_i, v_j \in T$ and $v_i \neq v_j$.

The exact calculation of the a priori π_i and the approximation of the joint probability ψ_{ij} can be obtained by using a smoothed version of the generative model introduced previously (LDA) (Blei et al., 2003), which makes use of Gibbs sampling (Griffiths et al., 2007). Once π_i and ψ_{ij} are known, the conditional probability ρ_{is} can be easily obtained through Bayes's rule (see Appendix A for further details on the probability computation).

Step 2: Roots Selection

The inputs of this step are the probability ρ_{is} and the value H, which is the number of special words (named roots) that will be selected to build the output set $\{r_i\}$.

We define a root as a special word of the vocabulary T with a high probability of occurring given that other words occurred in the set Ω_{fback}. Following this model, each word of the vocabulary can be a possible root. In our model, we consider a small number of roots, $H << |T|$, selecting them according to the highest occurrence probability.

The choice for the number H is made after a parameter tuning stage. As we will see below, when the number of documents is small, usually H is equal to 4 or 5.

To compute the probability of each root given the remaining words of the vocabulary, we introduce a graphical simplification. For each root, let us consider a directed acyclic graph (*dag*) that describes the relations between a root and the remaining words of the vocabulary (see Figure 3A). In particular, the probability of each root r_i given its parents

DOI: 10.1002/asi

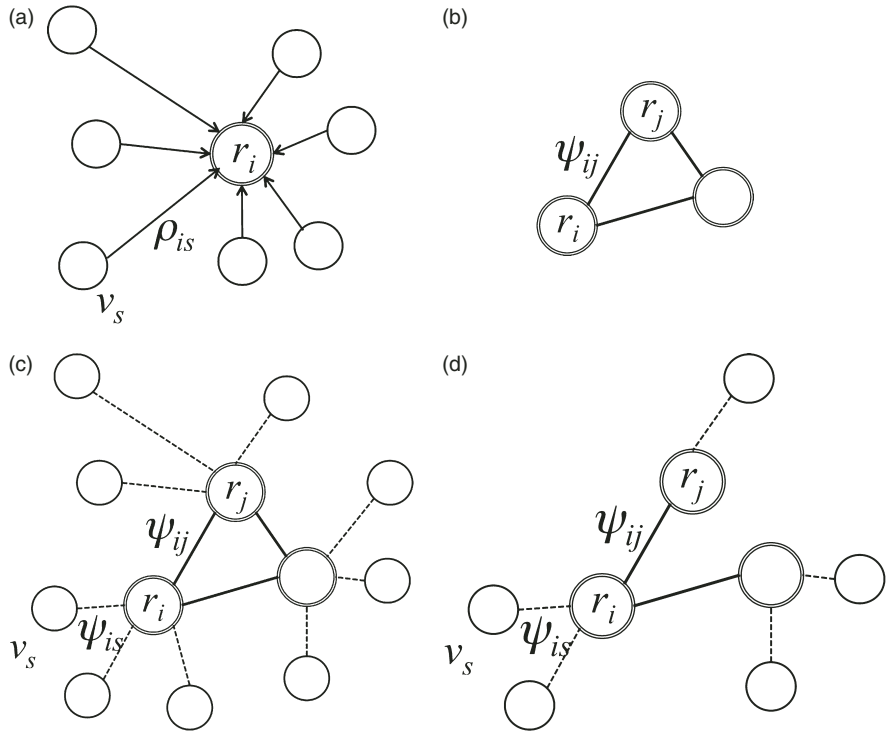

FIG. 3. Graphical representation of word associations. (a) *dag* between roots and words. (b) *ug* between roots. (c) Graph at step 3: *ug* of root-root and root-word. (d) Final graph after the optimization stage at step 4.

$(v_{par(r_i)})$ is computed by using the factorization property, as shown by Equation (2):

$$P(r_i|v_{par(r_i)}) = \prod_{s\neq i} P(r_i|v_s) = \prod_{s\neq i} \rho_{is} \tag{2}$$

Once the $P(r_i|v_{par(r_i)})\ \forall i$ are computed, we can select the best H roots $\{r_i\}$, by choosing those that have the highest probability.

Step 3: Root-Root and Root-Word Pairs Selection

The inputs of this step are the probabilities π_i, ψ_{ij}, and ρ_{is} and the roots $\{r_i\}$, while the outputs are two sets of probabilities describing root-root relations Ψ^{root}, and root-words relations Ψ_i^{words}, $\forall r_i$.

Once the H roots have been selected, we have H *dag*s. Starting from these *dag*s, we build undirected graphs (*ug*s) by considering the undirected relations between roots and words instead of directed relations. The *ug*s are described by the following probabilities: $\Psi_i^{words} = \{\psi_{is}\}_{s=1,\ldots,T,i\neq s}\ \ \forall i = 1, \ldots, H$.

Moreover, we build an undirected graph *ug* between the H roots (see Figure 3B). Such a graph describes all the possible associations between pairs of roots. The probabilities associated to this graph are: $\Psi^{roots} = \{\psi_{ij}\}_{i,j=1,\ldots,H}$, $i \neq j$.

Combining the *ug* between roots and the H *ug*s between roots and words, we obtain a preliminary version of the WWP (displayed in Figure 3C as a graph).

Step 4: Optimization Stage

The inputs of this step are the sets Ψ^{roots} and Ψ_i^{words}, $\forall i$, while the output is the vector $\mathbf{g} = (b_1, \ldots, b_{|G|})$ containing the weights of the $|G|$ word pairs $\{(v, u)_p\}$.

Note that if we choose H roots, we have H(H − 1)/2 root-root pairs, while the total number of possible root-word pairs is $(|T|(|T| - 1)/2 \times H$. As a consequence, the total number of pairs is $H(H - 1)/2 + (|T|(|T| - 1)/2) \times H$. For instance, for H = 4 and $|T| = 100$, we have 19,806 pairs.

The aim of the query expansion is to add some topic-related terms to the initial query. If we use the WWP to expand the query, we have to add 19,806 pairs of words that would be not efficient. For this reason, we perform an optimization stage to reduce the total number of pairs. We set a boundary condition for the optimization procedure by considering a maximum number of pairs equal to $|G|$.

The optimization stage, in addition to reduce the number of pairs, allows to neglect weakly related pairs according to a fitness function, which is discussed in Appendix B. In particular, our optimization strategy, given the number of roots H and the desired max number of pairs $|G|$, searches for a threshold λ and a set of thresholds $\{\mu_i\}_{i=1,\ldots,H}$ for cutting weak relations. More details are listed below:

1. λ: threshold that establishes the number of *root-root* pairs. A relations between two roots is relevant if $\psi_{ij} \geq \lambda$.
2. μ_i: threshold that establishes, given a root i, the number of *root-word* pairs. A relationship between the word v_s and the root r_i is relevant if $\psi_{is} \geq \mu_i$.

DOI: 10.1002/asi

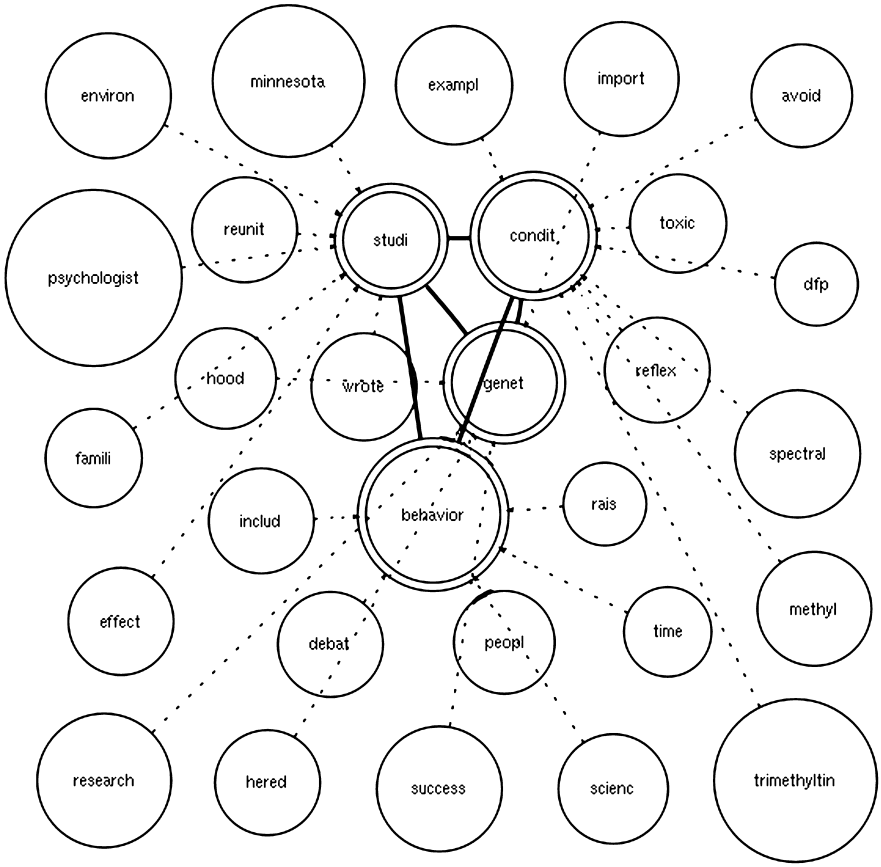

FIG. 4. Example of a WWP graph (Topic 402 TREC-8, "Behavioral genetics"). Double circles refer to *roots*.

TABLE 1. Fragment of tabular representation of a WWP for the example in Figure 4.

Term i	Term j	Weight
condit	behavior	0.029
studi	behavior	0.055
genet	condit	0.019
genet	studi	0.021
genet	behavior	0.005
studi	condit	0.027
includ	behavior	0.030
famili	studi	0.054

As a result of the optimization, we obtain the WWP that includes $|G|$ words pairs. A graphical depiction of the WWP is shown in Figure 3D. In practice, this graph is a reduced version of the graph shown in Figure 3C. Furthermore, in Figure 4, we show the WWP extracted from the topic 402 of TREC-8 "Behavioral genetics," while in Table 1 we show its tabular representation.

The final WWP is represented as a vector of weights $\mathbf{g} = (b_1, \ldots, b_{|G|})$ associated to the $|G|$ words pairs $\{(v,u)_p\}_{p=1}^{|G|}$. Each weight b_p represents the joint probability between two words, namely, $b_p = \psi_{ij}$.

From WWP Graph to the Expanded Query

Once the optimal WWP structure has been extracted from the feedback documents, it must be translated into an expanded query. This process, according to Figure 1, is called query reformulation and is carried out by considering a WWP graph (Figure 4) as a simple set of WWP (see tabular representation of a WWP in Table 1). In fact, at this stage, there is no more need to distinguish between roots and simple words, although this hierarchical distinction was fundamental for the structure building process. Note that the query reformulation process depends on the IR system considered.

There are several open-source libraries providing full-text search features. We have chosen Apache Lucene, Indri (Lemur Project), Terrier, and Zettair because they handle complex query expansions through custom Boolean weighted models. Because Boolean structured queries have shown to achieve poor performance, especially because of disjunctions handling (Kekäläinen & Järvelin, 1998), our basic idea was to translate the WWP plain representation (Table 1) through Boolean operators as follows:

- Original query is boosted of a default value (typically fixed to "1");
- Each word pair is translated as a binary AND between its two terms;

DOI: 10.1002/asi

- Each word pair is boosted of the relation weight;
- Expanded query is obtained as a Boolean OR between the original query and each WWP.

Considering Lucene as IR, the WWP plain representation (Table 1) is translated according to the Lucene Boolean model as follows:

```
(behavioral genetics)^1 OR (condit AND
behavior)^0.029
OR (studi AND behavior)^0.055. . .
```

Every word pair is searched with a Lucene boost factor chosen as the corresponding WWP weight ψ_{ij}, while the initial query is added with unitary boost factor (default). Lucene considers a Boolean query composed of nested queries made of Boolean clauses. It handles AND/OR operators by using "occur flags": MUST/SHOULD. These flags indicate that a document MUST match a nested query for matching the Boolean query or that a document SHOULD match a nested query for matching the Boolean query. The final score of a document, which matches a Boolean query, is based on the sum of the scores from all the matching Boolean clauses, multiplied by a special factor. This factor is the ratio of the number of matching Boolean clauses to the total number of Boolean clauses in the Boolean query.

A similiar translation was done for Indri, Terrier, and Zettair according to their own syntax (Ogilvie & Callan, 2002; Ounis et al., 2006; Strohman et al., 2005).

Experiments

The performance comparisons have been carried out testing the following FE/IR configurations:

- IR only. Unexpanded queries have been performed using first Lucene, Lemur, Terrier, and then Zettair as IR modules. Results obtained in these cases are referred to as baseline.
- FE(WWP) + IR. Our WWP-based FE method has been used to expand the initial query and feed all the IR modules considered. Both explicit and pseudorelevance feedback schemes have been used.
- FE(KLD) + IR. The KLD-based (Carpineto et al., 2001) FE method has been used to expand initial query and feed all the IR modules considered. Both explicit and pseudorelevance feedback schemes have been used.
- FE(RM3) + IR. The RM3-based (Abdul-Jaleel et al., 2004) FE method has been used to expand initial query and feed all the IR modules considered. Both explicit and pseudorelevance feedback schemes have been used.

Experimental Setup

We used the data sets from TREC disks 4 and 5 (minus the Congressional Records) and WT10G. Word stopping and word stemming (by using the Porters stemmer) with single-keyword indexing have been performed. The queries were generated from TREC topics 301–450 (TREC-6 through -8) and from topics 451–550 (TREC-9 and -10); see Table 2 for details on the test collections. We used the TREC-8 collection as a training set for the purpose of tuning the parameters of our method.

TABLE 2. Details of the collections used in the experiments.

Queries ID	No. of queries	Corpus
TREC678 301-450	150	TREC disk 4,5—CR
TREC910 451-550	100	WT10G

Query terms for each topic's initial search (baseline) have been obtained by parsing the title field of a topic (2.6 words per query, on average) to test a system's behavior in response to short base queries. For the baseline and first search task, needed for feedback document selection, the default similarity measures provided by Lucene, Lemur, Terrier, and Zettair have been used. Performance has been measured with TREC's suggested evaluation measures: mean average precision (MAP) and precision@10 (Manning et al., 2008). We have also performed a two-tailed paired t test with a confidence level of 95% to check for statistically significant differences between the proposed method and the others.

In the case of explicit feedback, we take the first mean (M) relevant documents from the result set returned by the system after the initial query. We simulate the behavior of a very *patient* user (highly motivated) (Keskustalo, Järvelin, & Pirkola, 2008) who is willing to browse up a significant number[5] of documents and select the M most relevant. In the case of PRF, we take the *top M* documents retrieved by the system in response to the the initial query. We do not take into account the relevance of each document because we assume that all the M documents are relevant. In this case, M is set to 10 in agreement with the observations of Carpineto et al. (2001)

Parameters Setting

The most important parameters involved in the computation of a WWP structure are the *number of roots H*, the *number of pairs* $|G|$, and the number of relevant documents M. It must be noted that the number M is set to 10 in the case of PRF. Parameters setting has been obtained by using the TREC-8 collection. Table 3 shows retrieval performances and computational times needed to build a WWP structure while varying the number of roots. Our choice was $H = 4$[6]

[5]We observed that if $M = 3$, a browsing window of maximum size $F = 100$ enables the user to find M documents that are certainly relevant according to TREC data sets' annotations, even when using Lucene to search for topics with few judged documents (worst case).

[6]Results have been obtained using an Intel Core 2 Duo 2,40 GHz PC with 4GB RAM with no other process running.

DOI: 10.1002/asi

because, in our view, it was the best trade-off between retrieval performances and computational times (as we can see from Table 3).

In order to choose the number M of relevant documents and the number $|G|$ of pairs, we evaluated the MAP achieved when varying at the same time M and $|G|$. As shown in Figure 5, highest MAP values (lightest gray) are obtained when the WWP has been built using three relevant documents and the number of pairs is set to 50 at least. The value of MAP shown in Figure 5 at (0,0) refers to the baseline (unexpanded query). The MAP values shown in Figure 5 have been obtained with Lucene, but a quite similar behavior has been observed for Lemur, Terrier, and Zettair.

TABLE 3. The number of roots H can be chosen as a trade-off between retrieval performances and computational times (our choice was H = 4).

H	MAP(%)	P@5(%)	Time(s)
2	26,00	72,00	3,98
3	27,95	73,60	4,6
4	29,09	76,00	6,06
5	29,17	76,24	9,5
6	30,04	73,60	12,04

Comparisons with Other Methods and Schemes

Performance analysis with explicit relevance feedback scheme. In Tables 4 and 5, we show results obtained comparing the WWP method with a KLD-based QE method, RM3, and the baseline. The tables report results obtained with all the considered IR modules. As we can see, WWP outperforms KLD, RM3, and baseline, especially for precision@10. However, because we did not remove the documents used for training from the test set, performance improvement is partially determined by the same training documents being reretrieved. This would make RF experiments not directly comparable with PRF (where this aspect

TABLE 4. Results comparison on TREC678 with explicit relevance feedback scheme.

IR model	Measure	Baseline	KLD	RM3	WWP
Lucene	#rel_ret	3672	5121	4221	5367
	map	0,1098	0,1691	0,1445	$0,1856^{+-+}$
	p@10	0,2607	0,2813	0,2781	0,2845
Lemur	#rel_ret	6517	7967	7884	8134
	map	0,1948	0,2448	0,2231	$0,2638^{+++}$
	p@10	0,3707	0,3913	0,3848	0,397
Terrier	#rel_ret	7285	8678	7756	8815
	map	0,2178	0,2765	0,2401	$0,2813^{+-+}$
	p@10	0,4291	0,4492	0,4413	0,4561
Zettair	#rel_ret	7301	8731	7793	8798
	map	0,2205	0,2731	0,2489	$0,2930^{+++}$
	p@10	0,4415	0,4671	0,4632	0,4685

Note. Superscripts + and – denote a statistically or a not statistically significant improvement of the proposed method over the baseline, KLD, and RM3, respectively.

TABLE 5. Results comparison on TREC910 with explicit relevance feedback scheme.

IR model	Measure	Baseline	KLD	RM3	WWP
Lucene	#rel_ret	1899	2384	2115	2550
	map	0,098	0,1082	0,1262	$0,1397^{++-}$
	p@10	0,1849	0,1984	0,2093	0,2163
Lemur	#rel_ret	3370	3710	3574	4353
	map	0,1737	0,1871	0,1979	$0,2186^{+++}$
	p@10	0,2630	0,2710	0,2921	0,2959
Terrier	#rel_ret	3768	4041	3942	4226
	map	0,1943	0,2153	0,2262	$0,2314^{++-}$
	p@10	0,3045	0,3115	0,3357	0,3382
Zettair	#rel_ret	3776	4066	3986	4229
	map	0,1967	0,2182	0,2234	$0,2398^{+++}$
	p@10	0,3132	0,3285	0,3490	0,3551

Note. Superscripts + and – denote a statistically or a not statistically significant improvement of the proposed method over the baseline, KLD, and RM3, respectively.

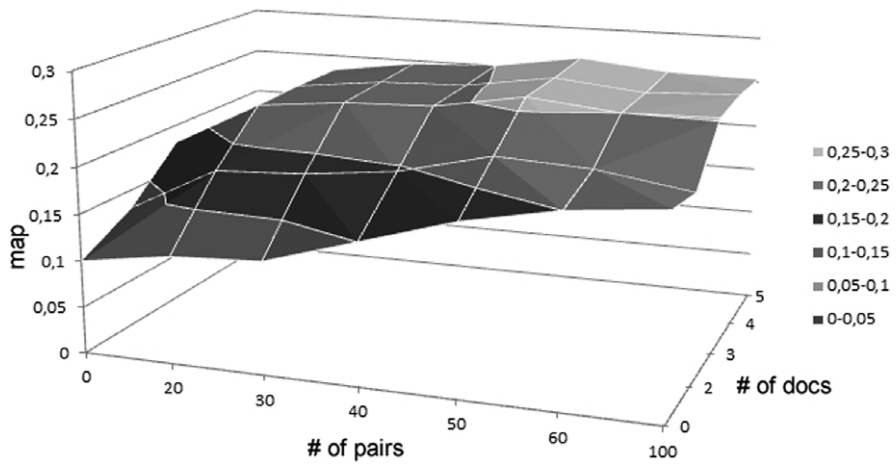

FIG. 5. WWP map performance achieved by Lucene by varying the number of pairs and number of relevant documents at the same time.

DOI: 10.1002/asi

TABLE 6. Results comparison on TREC678 with PRF scheme.

IR model	Measure	Baseline	KLD	RM3	WWP
Lucene	#rel_ret	3672	4941	3999	5041
	map	0,1098	0,1484	0,1209	$0,1617^{+++}$
	p@10	0,2607	0,266	0,2634	0,2739
Lemur	#rel_ret	6517	7786	6844	7886
	map	0,1948	0,2334	0,2059	$0,2467^{++-}$
	p@10	0,3707	0,376	0,3734	0,3839
Terrier	#rel_ret	7285	8554	7612	8654
	map	0,2178	0,2564	0,2289	$0,2697^{+++}$
	p@10	0,4291	0,4344	0,4318	0,4423
Zettair	#rel_ret	7301	8612	7703	8721
	map	0,2205	0,2601	0,2345	$0,2810^{+++}$
	p@10	0,4415	0,4567	0,4497	0,4598

Note. Superscripts + and – denote a statistically or a not statistically significant improvement of the proposed method over the baseline, KLD, and RM3, respectively.

TABLE 7. Results comparison on TREC910 with PRF scheme.

IR model	Measure	Baseline	KLD	RM3	WWP
Lucene	#rel_ret	1899	2301	2041	2416
	map	0,0980	0,0992	0,1108	$0,1169^{++-}$
	p@10	0,1849	0,1877	0,198	0,2049
Lemur	#rel_ret	3370	3626	3493	3779
	map	0,1737	0,1784	0,1887	$0,2018^{+++}$
	p@10	0,2630	0,2684	0,2807	0,2872
Terrier	#rel_ret	3768	3984	3886	4148
	map	0,1943	0,1997	0,2098	$0,2207^{++-}$
	p@10	0,3045	0,3098	0,3247	0,3310
Zettair	#rel_ret	3776	4011	3932	4180
	map	0,1967	0,2079	0,2128	$0,2260^{++-}$
	p@10	0,3132	0,3176	0,3413	0,3447

Note. Superscripts + and – denote a statistically or a not statistically significant improvement of the proposed method over the baseline, KLD, and RM3, respectively.

is not an issue) shown later. However, our main point is not to demonstrate that RF is better than PRF. An overall improvement of performance is more evident when using Zettair. The *t* test of statistical significance demonstrates that, in most of the cases, the improvements achieved by our method are likely not a result of chance.

Performance analysis with PRF. In Tables 6 and 7, we show the performance obtained by WWP, KLD, RM3, and baseline in the case of PRF.

WWP still outperforms the other methods and the *t* test of statistical significance demonstrates that, in most of the cases, the improvements achieved by our method are likely not a result of chance.

It is well known that the effectiveness of a pseudorelevance approach strictly depends on the quality of retrieved results in response to the initial query. Whatever the term selection method is (WWP, KLD, or RM3), the relevance of the first *M* documents of the result set can highly compromise the performance of the system.

TABLE 8. Expanded query building evaluation times.

Method	Time (sec/query)
KLD	0,027
RM3	0,018
WWP	0,040

As previously discussed, and shown in Tables 6 and 7, Lucene and Lemur have values of precision@10 lower than 0.3. This means that we have several nonrelevant documents among the first *M* documents returned by the system after the initial query. In contrast, Terrier and Zettair have values of precision@10 higher than 0.3. This makes the PRF more feasible in a real context.

Computational Times

Table 8 shows the average time (per topic) required to determine terms and weights for query expansion in order to make a comparison between the considered methods. The experimental stage was carried out using an Intel Core 2 Duo 2,40 GHz PC with 4GB RAM with no other user process running. We observe an increase in time when using WWP by a factor of 2.22, compared to RM3, and a factor of 1.48, compared to KLD. We found this drawback quite acceptable, given the retrieval effectiveness increase. Nevertheless, some efforts are currently being made to optimize the WWP extraction algorithm and lower computational times.

Conclusions

In this article, we investigated the use of a new term extraction method for QE based on a structured representation made of WWP. This structure is extracted from the set of documents obtained through the relevance feedback and then is added to the initial query. We showed that this structure can be easily employed in existing text retrieval systems through custom Boolean weighted models. Experiments have been conducted on a number of test collections (TREC-6, -7, -8, -9, and -10) and have demonstrated the effectiveness of this new structured representation with respect to other methods in the state of the art, such as KLD and RM3. Comparisons have been performed in both explicit and pseudorelevance scenarios. Tests of stastistical significance have been performed, confirming that, in most of the cases, the improvements achieved by this new method are not a result of chance. This article demonstrates that a structured feature representation (WWP) has a greater discriminating power than a feature vector made of weighted words.

The proposed approach computes the expanded queries considering only endogenous knowledge. It is well known that the use of external knowledge, for instance, WordNet, could clearly improve the accuracy of IR systems, and we will consider this integration in future work.

DOI: 10.1002/asi

References

Abdul-Jaleel, N., Allan, J., Croft, W.B., Diaz, F., Larkey, L., Li, X., et al. UMass at TREC 2004: Novelty and HARD. Technical report, DTIC document, 2004.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern information retrieval. New York: ACM.

Bai, J., & Nie, J.-Y. (2008). Adapting information retrieval to query contexts. Information Processing & Management, 44(6), 1901–1922. doi: 10.1016/j.ipm.2008.07.006; ISSN 0306-4573.

Bai, J., Song, D., Bruza, P., Nie, J.-Y., & Cao, G. (2005). Query expansion using term relationships in language models for information retrieval. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM '05 (pp. 688–695). New York: ACM. ISBN 1-59593-140-6, doi: 10.1145/1099554.1099725

Bhogal, J., MacFarlane, A., & Smith, P. (2007). A review of ontology based query expansion. Information Processing & Management, 43(4), 866–886. ISSN 0306-4573.

Blei, D.M., Ng, A.Y., & Jordan, M.I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3(1), 993–1022.

Callan, J., Croft, W.B., & Harding, S.M. (1992). The INQUERY retrieval system. In Proceedings of the Third International Conference on Database and Expert Systems Applications (pp. 78–83). Spain: Springer-Verlag.

Cao, G., Nie, J.-Y., & Bai, J. (2005). Integrating word relationships into language models. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05 (pp. 298–305). New York: ACM. ISBN 1-59593-034-5, doi: 10.1145/1076034.1076086

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. ACM Computing Surveys, 44(1), 1–50. doi: 10.1145/2071389.2071390; ISSN 0360-0300.

Carpineto, C., de Mori, R., Romano, G., & Bigi, B. (2001). An information-theoretic approach to automatic query expansion. ACM Transactions Information Systems, 19, 1–27. ISSN 1046-8188.

Chang, S.-K., Colace, F., Zhao, L., & Sun, Y. (2011). Processing continuous queries on sensor-based multimedia data streams by multimedia dependency analysis and ontological filtering. International Journal of Software Engineering and Knowledge Engineering, 21(8), 1169–1208.

Clarizia, F., Greco, L., & Napoletano, P. (2011). An adaptive optimisation method for automatic lightweight ontology extractions. In J. Filipe & J. Cordeiro (Eds.), Lecture notes in business information processing (pp. 357–371). Berlin/Heidelberg, Germany: Springer-Verlag. ISBN 978-3-540-28349-2.

Colace, F., De Santo, M., Greco, L., & Napoletano, P. (2013a). A query expansion method based on a weighted word pairs approach. In Proceedings of the 3rd Italian Information Retrieval (IIR) (Vol. 964, pp. 17–28). CEUR-WS: Pisa.

Colace, F., De Santo, M., Greco, L., & Napoletano, P. (2013b). Improving text retrieval accuracy by using a minimal relevance feedback. In Ana Fred, J. Dietz, K. Liu, & Jose Filipe (Eds), Knowledge discovery, knowledge engineering and knowledge management (Vol. 348, pp. 126–140). Berlin Heidelberg: Springer.

Colace, F., De Santo, M., Greco, L., & Napoletano, P. (2014). Text classification using a few labeled examples. Computers in Human Behavior, 30(1), 689–697.

Collins-Thompson, K., & Callan, J. (2005). Query expansion using random walk models. In Proceedings of the 14th ACM International Conference on Information and knowledge management, *CIKM '05* (pp. 704–711). New York: ACM. ISBN 1-59593-140-6.

Dumais, S., Joachims, T., Bharat, K., & Weigend, A. (2003). SIGIR 2003 workshop report: Implicit measures of user interests and preferences. SIGIR Forum, 37(2), 50–54. ISSN 0163-5840.

Efthimiadis, E.N. (1996). Query expansion. In M.E. Williams (Ed.), Annual Review of Information Systems and technology (pp. 121–187). Melford, NJ, USA: Information Today.

Griffiths, T.L., Steyvers, M., & Tenenbaum, J.B. (2007). Topics in semantic representation. Psychological Review, 114(2), 211–244.

Jansen, B.J., Spink, A., & Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the Web. Information Processing & Management, 36(2), 207–227. ISSN 0306-4573.

Jansen, B.J., Booth, D.L., & Spink, A. (2008). Determining the informational, navigational, and transactional intent of Web queries. Information Processing & Management, 44(3), 1251–1266. ISSN 0306-4573.

Jing, Y., & Croft, W.B. (1994). An association thesaurus for information retrieval. In RIAO 94 Conference Proceedings (pp. 146–160). New York: CID.

Kekäläinen, J., & Järvelin, K. (1998). The impact of query structure and query expansion on retrieval performance. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 130–137). Melbourne: ACM.

Keskustalo, H., Järvelin, K., & Pirkola, A. (2008). Evaluating the effectiveness of relevance feedback based on a user simulation model: Effects of a user scenario on cumulated gain value. Information Retrieval, 11(5), 209–228.

Ko, Y., & Seo, J. (2009). Text classification from unlabeled documents with bootstrapping and feature projection techniques. Information Processing & Management, 45(1), 70–83. ISSN 0306-4573.

Lang, H., Metzler, D., Wang, B., & Li, J.-T. (2010). Improved latent concept expansion using hierarchical Markov random fields. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10 (pp. 249–258). New York: ACM. ISBN 978-1-4503-0099-5.

Lavrenko, V., & Croft, W.B. (2001). Relevance based language models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01 (pp. 120–127). New York: ACM. ISBN 1-58113-331-6; doi: 10.1145/383952.383972. Retrieved from http://doi.acm.org/10.1145/383952.383972

Manning, C.D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge: Cambridge University.

Metzler, D., & Croft, W.B. (2007). Latent concept expansion using Markov random fields. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07 (pp. 311–318). New York: ACM. ISBN 978-1-59593-597-7; doi: 10.1145/1277741.1277796

Na, S.-H., Kang, I.-S., Roh, J.-E., & Lee, J.-H. (2005). An empirical study of query expansion and cluster-based retrieval in language modeling approach. In G. Lee, A. Yamada, H. Meng, & S. Myaeng (Eds.), Proceedings of the 2nd Asian Information Retrieval Symposium, Lecture Notes in Computer Science (Vol. 3689, pp. 274–287). Berlin/Heidelberg, Germany: Springer. ISBN 978-3-540-29186-2.

Ogilvie, P., & Callan, J. (2002). Experiments using the Lemur toolkit. In Proceedings of the Tenth Text Retrieval Conference (TREC-10) (pp. 103–108). Gaitersburgh, Maryland, USA: NIST.

Okabe, M., & Yamada, S. (2007). Semisupervised query expansion with minimal feedback. IEEE Transactions on Knowledge and Data Engineering, 19(1), 1585–1589. doi: http://doi.ieeecomputersociety.org/10.1109/TKDE.2007.190646; ISSN 1041-4347.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., & Lioma, C. (2006). Terrier: A high performance and scalable information retrieval platform. In Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006) (pp. 18–25). Seattle, Washington, USA: ACM SIGIR.

Park, L.A.F., & Ramamohanarao, K. (2009). An analysis of latent semantic term self-correlation. ACM Transactions Information Systems, 27(2), 1–8, 8, 35. doi: 10.1145/1462198.1462200; ISSN 1046-8188.

Pinto, F.J., Farina Martinez, A., & Perez-Sanjulian, C.F. (2008). Joining automatic query expansion based on thesaurus and word sense disambiguation using wordnet. International Journal of Computer Applications Technology, 33(4), 271–279. doi: 10.1504/IJCAT.2008.022422; ISSN 0952-8091.

Qiu, Y., & Frei, H.-P. (1993). Concept based query expansion. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '93 (pp. 160–169). New York: ACM. ISBN 0-89791-605-0,

doi: 10.1145/160688.160713; Retrieved from http://doi.acm.org/10.1145/160688.160713

Strohman, T., Metzler, D., Turtle, H., & Croft, W.B. (2005). Indri: A language-model based search engine for complex queries. Technical report. In Proceedings of the International Conference on Intelligent Analysis (pp. 2–6). McLean, VA, USA: ICIA.

Xu, J., & Croft, W.B. (1996). Query expansion using local and global document analysis. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '96 (pp. 4–11). New York: ACM. ISBN 0-89791-792-8, doi: 10.1145/243199.243202; Retrieved from http://doi.acm.org/10.1145/243199.243202

Zhang, J., Deng, B., & Li, X. (2009). Concept based query expansion using WordNet. In Advanced Science and Technology, 2009. AST 09. International e-Conference (pp. 52–55). Dajeon: IEEE. doi: 10.1109/AST.2009.24

Appendix A. Priori and Joint Probability Computation

The LDA theory, introduced by Blei et al. (2003; Griffiths et al., 2007), considers a semantic representation in which a document is represented in terms of a set of probabilistic topics z. More formally, let us consider a word v_i of a document $\mathbf{d}_m$ as a random variable on the vocabulary T and z as a random variable representing one of the topics between $\{1, \ldots, K\}$. To obtain a word, the model considers three parameters assigned: α, η, and the number of topics K. Given these parameters, the model chooses θ_m through $P(\theta|\alpha) \sim Dirichlet(\alpha)$, the topic k through $P(z|\theta_m) \sim Multinomial(\theta_m)$ and $\beta_k \sim Dirichlet(\eta)$. Finally, the distribution of each word given a topic is $P(u_m|z, \beta_z) \sim Multinomial(\beta_z)$. The output obtained by performing Gibbs sampling on a set of documents Ω_{fback} consists of two matrixes:

1. The words-topics matrix Φ that contains $|T| \times K$ elements representing the probability that a word v_i of the vocabulary is assigned to topic k: $P(u = v_i|z = k, \beta_k)$.
2. The topics-documents matrix Θ that contains $K \times |\Omega_{fback}|$ elements representing the probability that a topic k is assigned to some word token within a document $\mathbf{d}_m$: $P(z = k|\theta_m)$.

The probability distribution of a word u_m within a document $\mathbf{d}_m$ of the corpus can be then obtained as shown in Equation (A1):

$$P(u_m) = \sum_{k=1}^{K} P(u_m|z = k, \beta_k)P(z = k|\theta_m). \quad \text{(A1)}$$

In the same way, the joint probability between two words u_m and y_m of a document $\mathbf{d}_m$ of the corpus can be obtained by assuming that each pair of words is represented in terms of a set of topics z and then as shown by Equation (A2):

$$P(u_m, y_m) = \sum_{k=1}^{K} P(u_m, y_m|z = k, \beta_k)P(z = k|\theta_m) \quad \text{(A2)}$$

Note that the exact calculation of Equation (A2) depends on the exact calculation of $P(u_m, y_m|z = k, \beta_k)$ that cannot be directly obtained through LDA. If we assume that words in a document are conditionally independent given a topic, an approximation for Equation (4) can be written as Equation (A3):

$$P(u_m, y_m) \simeq \sum_{k=1}^{K} P(u_m|z = k, \beta_k)P(y_m|z = k, \beta_k)P(z = k|\theta_m). \quad \text{(A3)}$$

Moreover, Equation (A1) gives the probability distribution of a word u_m within a document $\mathbf{d}_m$ of the corpus. To obtain the probability distribution of a word u independently of the document, we need to sum over the entire corpus, as shown in Equation (A4):

$$P(u) = \sum_{m=1}^{M} P(u_m)\delta_m \quad \text{(A4)}$$

where δ_m is the previous probability for each document ($\sum_{m=1}^{|\Omega fback|} \delta_m = 1$). In the same way, if we consider the joint probability distribution of two words u and y, we obtain Equation (A5):

$$P(u, y) = \sum_{m=1}^{M} P(u_m, y_v)\delta_m \quad \text{(A5)}$$

Concluding, once we have $P(u)$ and $P(u, y)$, we can compute $P(v_i) = P(u = v_i)$ and $P(v_i, v_j) = P(u = v_i, y = v_j)$, $\forall i, j \in \{1, \ldots, |T|\}$.

Appendix B. Optimization Stage

Given the maximum number of roots H and the maximum number of pairs $|G|$, several WWP structures $\mathbf{g}_t$ can be obtained by varying the parameters $\Lambda_t = (\tau, \mu)_t$. To find the best parameters Λ_t, we perform an optimization procedure that uses a scoring function and a searching strategy. As we have previously seen, a $\mathbf{g}_t$ is a vector of features $\mathbf{g}_t = (b_{1t}, \ldots, b_{|G|t})$ in the space G of the words pairs. Each document of the set Ω_{fback} can be represented as a vector $\mathbf{d}_m = (w_{1m}, \ldots, w_{|G|m})$ in the space G. A possible scoring function is the cosine similarity between these two vectors, as shown by Equation (B1):

$$S(\mathbf{g}_t, \mathbf{d}_m) = \frac{\sum_{n=1}^{|G|} b_{nt} \cdot w_{nm}}{\sqrt{\sum_{n=1}^{|G|} b_{nt}^2} \cdot \sqrt{\sum_{n=1}^{|G|} w_{nm}^2}} \quad \text{(B1)}$$

and thus the optimization procedure would consist in searching for the best set of parameters Λ_t such that the cosine similarity is maximized $\forall \mathbf{d}_m$. Therefore, the best $\mathbf{g}_t$ for the set of documents Ω_{fback} is the one that produces the maximum score attainable for each document when used to rank Ω_{fback} documents. Because a score for each document $\mathbf{d}_m$ is obtained, we have:

$$\mathbf{S}_t = \left\{ S(\mathbf{g}_t, \mathbf{d}_1), \cdots, S\left(\mathbf{g}_t, \mathbf{d}_{|\Omega_{fback}|}\right) \right\},$$

where each score depends on the specific set $\Lambda_t = (\tau, \mu)_t$. To find the best Λ_t, we can maximize the score value for each

document, which means that we are looking for the graph that best describes each document of the repository from which it has been extracted. This optimization procedure needs to maximize all $|\Omega_{fback}|$ elements of $\mathbf{S}_t$ at the same time. Alternatively, in order to reduce the number of the objectives being optimized, we can, at the same time, maximize the mean value of the scores and minimize their standard deviation, which turns a multi-objective problem into a one-objective one. Finally, the *Fitness* (F) will be:

$$F(\Lambda_t) = E[\mathbf{S}_t] - \sigma[\mathbf{S}_t],$$

where E is the mean value of all the elements of $\mathbf{S}_t$ and σ is the standard deviation. By summing up, the best parameters are such that Equation (B2):

$$\Lambda^* = \arg\max_t \{F(\Lambda_t)\} \tag{B2}$$

As discussed earlier, the space of possible solutions could grow exponentially. For this reason, we considered $|G| \leq 50$.

Moreover, because the number of possible values of Λ_t is, in theory, infinite, we clustered each set of τ and μ_s separately by using the *k-means* algorithm. In practice, we grouped all the values of ψ_{ij} and ρ_{is} in a few number of clusters. In this way, the optimum solution can be exactly obtained after the exploration of all the possible values of ψ_{ij} and ρ_{is} used as thresholds.

DOI: 10.1002/asi