Information Processing and Management 51 (2015) 179–193

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Weighted Word Pairs for query expansion ☆

Francesco Colace [a], Massimo De Santo [a], Luca Greco [a,*], Paolo Napoletano [b]

[a] DIEM, University of Salerno, Fisciano, Italy
[b] DISCo, University of Milano-Bicocca, Italy

ARTICLE INFO

Article history:
Received 25 June 2013
Received in revised form 4 July 2014
Accepted 11 July 2014
Available online 7 August 2014

Keywords:
Text retrieval
Query expansion
Explicit relevance feedback
Pseudo-relevance feedback
Probabilistic Topic Model

ABSTRACT

This paper proposes a novel query expansion method to improve accuracy of text retrieval systems. Our method makes use of a minimal relevance feedback to expand the initial query with a structured representation composed of weighted pairs of words. Such a structure is obtained from the relevance feedback through a method for pairs of words selection based on the Probabilistic Topic Model. We compared our method with other baseline query expansion schemes and methods. Evaluations performed on TREC-8 demonstrated the effectiveness of the proposed method with respect to the baseline.

1. Introduction

Most retrieval systems show relative weaknesses in retrieving relevant documents, especially when few keywords are used to model user information needs. Information retrieval models, that have been proposed through the years, often rely on the *bag of words* model for document and query representation and can be grouped into three main categories: set-theoretic (including boolean), algebraic and probabilistic models (Christopher, Manning, & Schtze, 2008; Baeza-Yates & Ribeiro-Neto, 1999). It is well known that the "bag of words" model assumes both documents and queries representable as feature vectors. The elements of such vectors can indicate the presence (or absence) of a word or take into account its occurrence frequency, but the information about the position of that word within the document is completely lost (Christopher et al., 2008); then, the elements of the vector are simply weights computed in different ways. In this context, the relevance of a document to a query can be measured as the distance between the corresponding vector representations in the space of features.

It has been found that common users are used to perform short queries, 2 or 3 words on average (Jansen, Spink, & Saracevic, 2000; Jansen, Booth, & Spink, 2008). Unfortunately, the shortness of a query can cause common information retrieval systems failures due to the inherent ambiguity of language (polysemy, etc.). Since most text retrieval systems relying on a term-frequency based index generally suffer from low precision (or low quality document retrieval), a typical solution adopted to reduce this query/document mismatch is expanding the initial 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): this strategy is often referred as *query expansion*.

☆ The authors contributed equally to this work.

* Corresponding author.
E-mail address: lgreco@unisa.it (L. Greco).

http://dx.doi.org/10.1016/j.ipm.2014.07.004
0306-4573/

In this work we propose a query expansion method that automatically extracts a set of Weighted Word Pairs from a set of topic-related documents provided by the relevance feedback. Such a structured set of terms is obtained by using a method of *term extraction* previously investigated in Colace, De Santo, Greco, and Napoletano (2013, 2014), Clarizia, Greco, and Napoletano (2011) and based on the *Latent Dirichlet Allocation* model (Blei, Ng, & Jordan, 2003) implemented as the *Probabilistic Topic Model* (Griffiths, Steyvers, & Tenenbaum, 2007).

Evaluation has been conducted on TREC-8 repository. We compared the proposed Weighted Word Pairs (WWP) with a method for term extraction based on the Kullback Leibler divergency (Carpineto et al., 2001). Our approach achieves overall better performances and demonstrates that a structured feature representation has a greater discriminating power than a feature vector made of weighted words.

2. Problem formulation

According to the Information Retrieval (IR) theory, the representation of queries and documents is based on the *Vector Space Model* (Christopher et al., 2008): a document or query is a vector of weighted words belonging to a vocabulary $\mathcal{T}$:

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

Each weight w_n is such that $0 \leqslant w_n \leqslant 1$ and represents how much the term t_n contributes to the semantics of the document $\mathbf{d}$ (in the same way for $\mathbf{q}$). 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:

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

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

2.1. Query expansion by relevance feedback

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

A general query expansion framework is a modular system including one or several instances, properly chained, of the following modules: Information Retrieval (IR), Feedback (F), Feature Extraction (FE), Query Reformulation (QR).

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

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 (pseudo-relevance) assigned. In 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 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}.

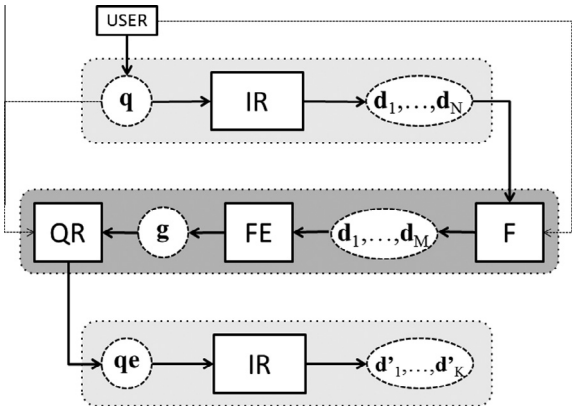

Fig. 1. General framework for query expansion.

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 Weighted Word Pairs proposed in this paper. The Query Reformulation (QR) module adapts the resulting set of features **g** in order to 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 $\Omega_{res} = \{\mathbf{d}'_1, \ldots, \mathbf{d}'_K\}$ is retrieved.

The query expansion framework described above is quite general. We can use any of the existing IR systems, as well as any of the existing methods of feature extraction, etc. According to this framework, we can make objective comparisons between different system configurations. In this paper we propose a new method of query expansion that use a set of structured features extracted from a minimal relevance feedback. We considered two different open source IR systems: Apache Lucene (Foundation, 2011) that supports structured query based on a weighted boolean model, and the Indri Lemur Toolkit (Ogilvie et al., 2002) that supports an extended set of probabilistic structured query operators based on *Inquery*. Moreover we compared our feature extraction method with one in the state of the art by considering both the explicit and the pseudo-relevance feedback schemes.

3. Background and related works

3.1. Query expansion techniques

The idea of taking advantage of additional knowledge to retrieve relevant documents has been largely discussed in the literature, where manual, interactive and automatic techniques have been proposed (Efthimiadis, 1996; Christopher et al., 2008; Baeza-Yates & Ribeiro-Neto, 1999; Carpineto & Romano, 2012; 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. ontology, WordNet, data mining) or *endogenous* knowledge (i.e. extracted only from the documents contained in the collection) (Bhogal, Macfarlane, & Smith, 2007; Christopher et al., 2008).

In this work we focus mainly on those query expansion techniques which make use of the *relevance feedback*. We can distinguish between three types of procedures for relevance assignment: explicit feedback, implicit feedback, and pseudo feedback (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. Pseudo relevance feedback (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.

Since human labeling task is enormously boring and time consuming (Ko & Seo, 2009), most existing methods make use of pseudo relevance feedback. Nevertheless, fully automatic methods can exhibit low performance when the initial query is intrinsically ambiguous. As a consequence, in recent years, some hybrid techniques have been developed which take into account a minimal explicit human feedback (Okabe & Yamada, 2007; Dumais, Joachims, Bharat, & Weigend, 2003) and use it to automatically identify other topic related documents. Such methods uses 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.

3.2. Term extraction techniques

Whatever the technique that selects the set of documents representing the feedback, the expanded terms are usually computed by making use of well known approaches for term selection as Rocchio, Robertson, CHI-Square, Kullback–Lieber, etc. (Carpineto et al., 2001; Carpineto & Romano, 2012; Cao, Nie, Gao, & Robertson, 2008). In this case the reformulated query consists in a simple (sometimes weighted) list of words.

Although such term selection 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 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).

Others propose methods based on language modeling to integrate several contextual factors in order to adapt document ranking to the specific query context (Bai & Nie, 2008) or to integrate term relationships Bai, Song, Bruza, Nie, and Cao, 2005. The latent semantic analysis (LSA) has been extensively used in information retrieval, especially for term correlations computing Park and Ramamohanarao, 2009.

Furthermore, several existing term selection methods use language models combined with exogenous knowledge, like thesaurus(Cao, Nie, & Bai, 2005), wordnet (Zhang, Deng, & Li, 2009; Pinto, Martinez, & Perez-Sanjulian, 2008) or ontology (Bhogal, Macfarlane, & Smith, 2007).

4. The proposed Weighted Word Pairs extraction method

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

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

containing the weights of the $|\mathcal{G}|$ word pairs $\{(v,u)_p\}$. The set $\mathcal{G}$ is the vocabulary of word pairs. The entire extraction process is divided into 4 steps, and it is showed in Fig. 2.

4.1. 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 $\mathcal{T}$. The outputs of this step are:

1. the a priori probability that a word v_i occurs in $\Omega_{fback} : \pi_i = P(v_i),\ \forall v_i \in \mathcal{T}$;
2. the conditional probability that a word v_i occurs in Ω_{fback} given that another word v_s occurred in $\Omega_{fback} : \rho_{is} = P(v_i|v_s),\ \forall v_i, v_s \in \mathcal{T}$ and $v_i \neq v_s$;
3. the joint probability that a pair of words, v_i and v_j, occurs at the same time in $\Omega_{fback} : \psi_{ij} = P(v_i, v_j),\ \forall v_i, v_j \in \mathcal{T}$ 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 in Blei et al. (2003) called Latent Dirichlet Allocation (LDA), 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 the Bayes' rule.

4.1.1. Probabilities computation through the Topic Model

The Latent Dirichlet Allocation (LDA) theory introduced by Blei et al. (2003) and 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 $\mathcal{T}$ and z as a random variable representing one of the topic 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 $|\mathcal{T}| \times K$ 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:

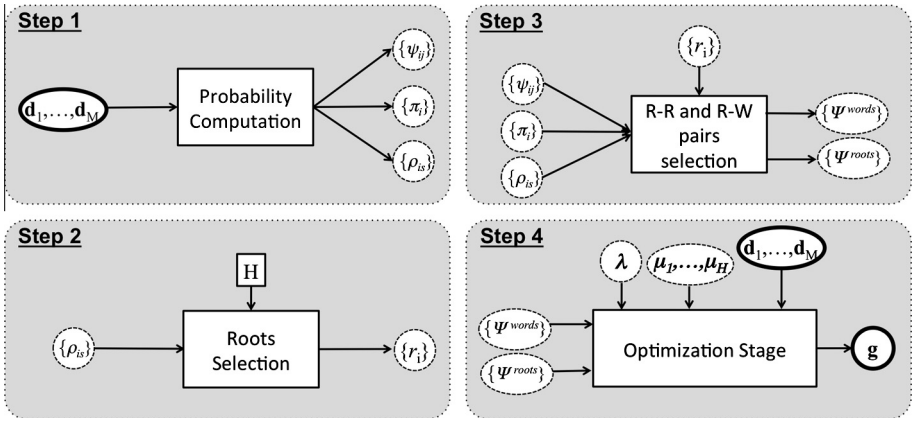

Fig. 2. Steps of the proposed feature extraction method.

$$P(u_m) = \sum_{k=1}^{K} P(u_m|z=k,\beta_k)P(z=k|\theta_m). \tag{2}$$

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:

$$P(u_m, y_m) = \sum_{k=1}^{K} P(u_m, y_m|z=k,\beta_k)P(z=k|\theta_m). \tag{3}$$

Note that the exact calculation of Eq. (3) 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 Eq. (3) can be written as:

$$P(u_m, y_m) \simeq \sum_{k=1}^{K} P(u_m|z=k,\beta_k)P(y_m|z=k,\beta_k)P(z=k|\theta_m). \tag{4}$$

Moreover, Eq. (2) 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:

$$P(u) = \sum_{m=1}^{M} P(u_m)\delta_m, \tag{5}$$

where δ_m is the prior 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:

$$P(u, y) = \sum_{m=1}^{M} P(u_m, y_v)\delta_m. \tag{6}$$

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

4.2. 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 $\mathcal{T}$ having a high probability to occur 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, $H \ll |\mathcal{T}|$ 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 later in the paper, 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 r_i and the remaining words ($v_{par(r_i)}$) of the vocabulary, see Fig. 3a. Then, the probability of each root can be computed by using the factorization property:

$$P(r_i|v_{par(r_i)}) = \prod_{s\neq i} P(r_i|v_s) = \prod_{s\neq i}\rho_{is}. \tag{7}$$

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.

4.3. Step 3: root-root and root-word pairs selection

The inputs of this step are the probabilities π_i, ψ_{ij}, 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 *dags*. Starting from these *dags* we build undirected graphs (*ugs*) by considering the undirected relations between roots and words instead of directed relations. The *ugs* are described by the following probabilities: $\Psi_i^{words} = \{\psi_{is}\}_{s=1,\ldots,\mathcal{T},i\neq s}\ \forall i = 1,\ldots,H$.

Moreover, we build an undirected graph *ug* between the H roots, see Fig. 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 *ugs* between roots and words, we obtain a preliminary version of the weighted words pairs, that is displayed in Fig. 3c. as a graph.

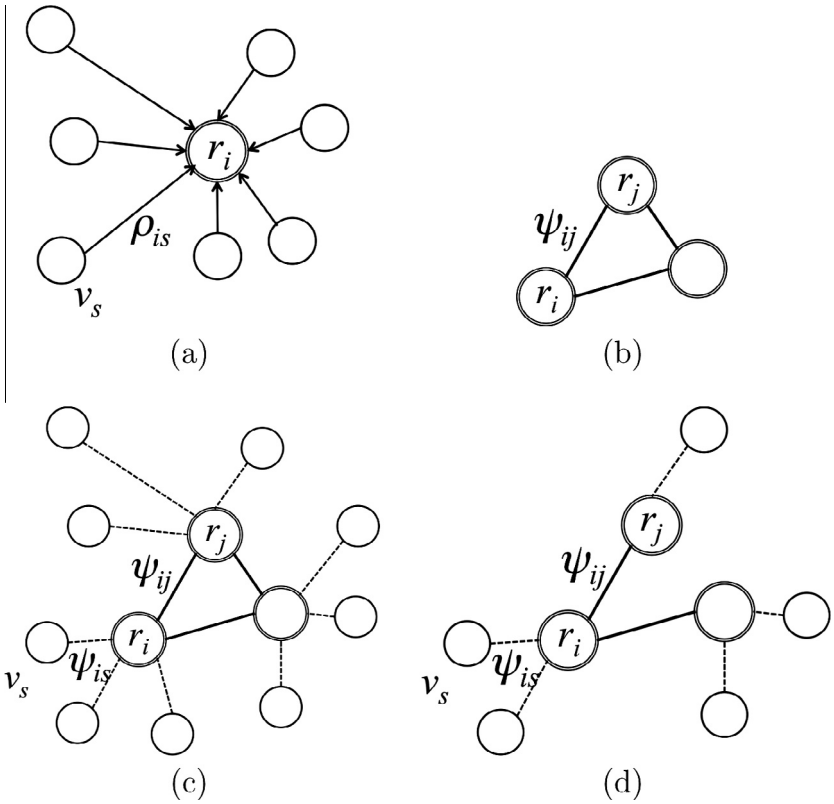

Fig. 3. Graphical representation of words associations. (a) *dag* between roots and words. (b) *ug* between roots. (c) Graph at step 3: *ug* of root-root and root-word. (d) Final graph after the optimization stage at step 4.

4.4. Step 4: optimization stage

The inputs of this step are the sets Ψ^{roots} and $\Psi_i^{words}, \forall i$, while the output is the vector $\mathbf{g} = \{b_1, \ldots, b_{|\mathcal{G}|}\}$ containing the weights of the $|\mathcal{G}|$ 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 $(|\mathcal{T}|(|\mathcal{T}|-1)/2) \times H$. As a consequence, the total number of pairs is $H(H-1)/2+(|\mathcal{T}|(|\mathcal{T}|-1)/2) \times H$. For instance for $H = 4$ and $|\mathcal{T}| = 100$, we have 19806 pairs.

The scope of the query expansion is to add some topic related terms to the initial query. If we use the Weighted Word Pairs 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 of the optimization procedure by considering a maximum number of pairs equal to $|\mathcal{G}|$.

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 A. In particular, our optimization strategy, given the number of roots H and the desired max number of pairs $|\mathcal{G}|$, search for a threshold λ and a set of thresholds $\{\mu_i\}_{i=1,\ldots,H}$ for cutting weak relations. More in details:

1. λ: threshold that establishes the number of *root-root* pairs. A relations between two roots is relevant if $\psi_{ij} \geqslant \lambda$.
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 $\psi_{is} \geqslant \mu_i$.

Once λ and $\{\mu_i\}_{i=1,\ldots,H}$ are known, the final WWP graph is obtained by selecting the right pairs from Ψ^{roots} and Ψ_i^{words}, $\forall i$. The graph is composed of $|\mathcal{G}|$ words pairs and is represented as a vector of weights $\mathbf{g} = \{b_1, \ldots, b_{|\mathcal{G}|}\}$ associated to the $|\mathcal{G}|$ words pairs $\{(v,u)_p\}_{p=1}^{|\mathcal{G}|}$. Each weight b_p represents the joint probability between two words, namely $b_p = \psi_{ij}$.

A graphical depiction of the WWP is showed in Fig. 3d. In practice this graph is a reduced version of the graph showed in Fig. 3c. Furthermore, in Fig. 4 we show the WWP extracted from the topic 402 of TREC-8 "Behavioral genetics", while in Table 1 we show its tabular representation. Double circle represents roots while single circle represents simple words. Dotted lines stand for relations between roots and words, while solid lines stand for relations between roots.

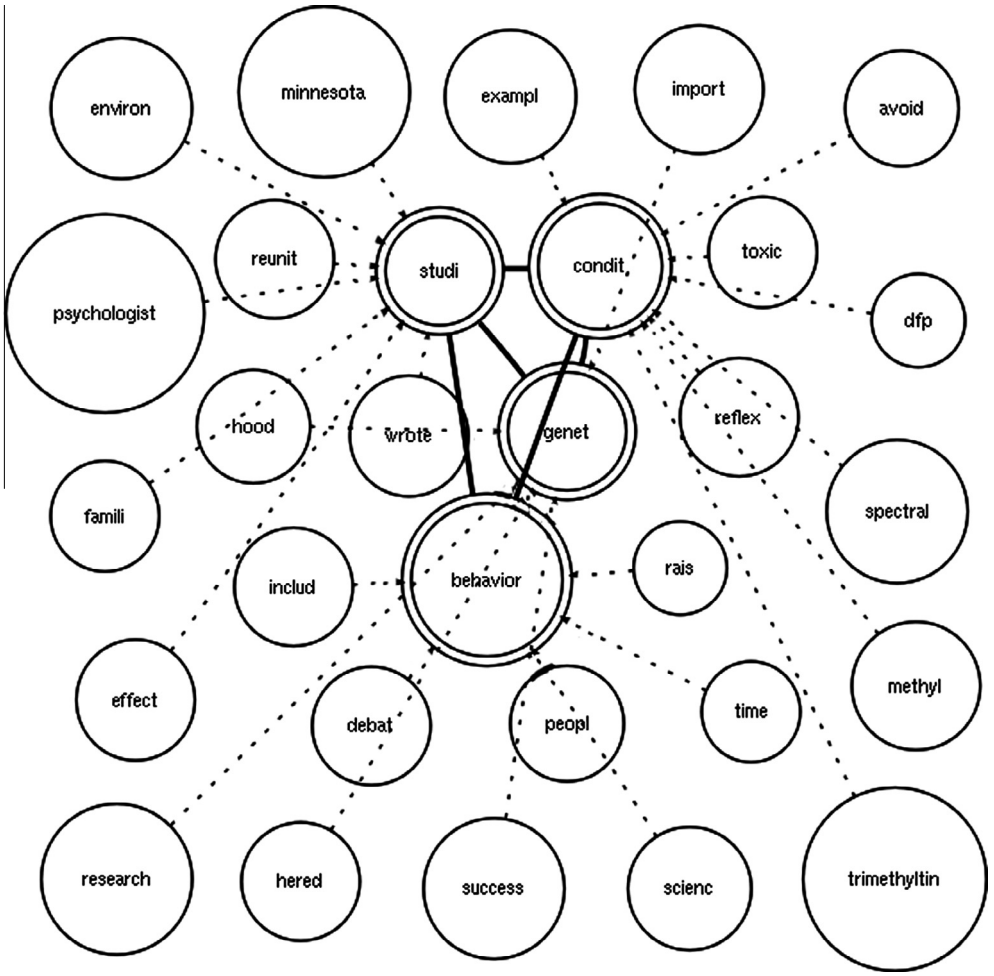

Fig. 4. Example of a *Weighted Word Pairs* graph (Topic 402 TREC-8, "Behavioral genetics").

In Algorithm 1 we show the pseudo-code of the procedure for graph building. Each function recalled in the pseudo-code has been described above.

Algorithm 1. WWP graph $\mathbf{g} = \{b_1, \ldots, b_{|\mathcal{G}|}\}$ building

Require: Data, M documents $\Omega_{fback} = \{\mathbf{d}_1, \ldots, \mathbf{d}_M\}$ on the vocabulary $\mathcal{T}$
Require: Parameters, $|\mathcal{G}|, H, \alpha, \eta, K$
$\{\pi_i, \psi_{ij}, \rho_{is}\}_{\forall i,j,s \in \mathcal{T}} = ProbabilityComputation(\{\mathbf{d}_1, \ldots, \mathbf{d}_M\}, \alpha, \eta, K)$
$\{r_h\}_{h=1}^{H} = RootSelection(\{\rho_{is}\}_{\forall i,s \in \mathcal{T}}, H)$
for $h = 1$ to H **do**
$\Psi_h^{words} = RootWordsPairsSelection(r_h, \{\psi_{hj}\}_{\forall j \in \mathcal{T}})$
end for
$\Psi^{roots} = RootRootPairsSelection(\{r_h\}_{h=1}^{H}, \{\psi_{ij}\}_{\forall i,j \in \mathcal{T}})$
$(\lambda, \{\mu_i\}_{i=1}^{H}) = Optimization(\{\mathbf{d}_1, \ldots, \mathbf{d}_M\}, \Psi^{roots}, \{\Psi_h^{words}\}_{h=1}^{H}, |\mathcal{G}|)$
$\mathbf{g} = GraphSelection(\Psi^{roots}, \{\Psi_h^{words}\}_{h=1}^{H}, \lambda, \{\mu_i\}_{i=1}^{H})$

Table 1
Fragment of tabular representation of a WWP for the example in Fig. 4.

Word *i*	Word *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

4.5. 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 Fig. 1, is called *query reformulation* and is carried out by considering a WWP graph (Fig. 4) as a simple set of Weighted Word Pairs (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 (Foundation, 2011) and Lemur Project (Ogilvie et al., 2002) since they handle complex query expansions through custom boolean weighted models. Considering Lucene as IR (Foundation, 2011), the WWP plain representation (Table 1) is translated according to 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).

When Lemur is used as IR module, WWP plain representation is translated into an expanded query using Indri query language as follows:

```
#weight(0.50 #combine(behavioral genetics) 0.50 #weight(0.029 #band(condit behavior) 0.055 #band
(studi behavior) ....
```

Lemur toolkit (Ogilvie et al., 2002) provides belief operators which allow to combine beliefs (scores) about terms, phrases, etc. There are both unweighted and weighted belief operators. With the weighted operators, weights can be assigned to certain expressions in order to control how much of an impact each expression within the query has on the final score.

5. Experiments

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

- **IR only**. Unexpanded queries has been performed using first Lucene and then Lemur as IR modules. Results obtained in these cases are referred as baseline.
- **FE(WWP) + IR**. Our WWP-based feature extraction method has been used to expand the initial query and feed Lucene and Lemur IR modules. Both explicit and pseudo-relevance relevance feedback schemes have been used.
- **FE(Random) + IR**. A WWP with random weights has been used to expand the initial query and feed Lucene and Lemur IR modules. Explicit feedback scheme has been used.
- **FE(KLD) + IR**. Kullback Leibler Divergency (Carpineto et al., 2001) based feature extraction method has been used to expand initial query and feed Lucene and Lemur IR modules. Both explicit and pseudo-relevance feedback schemes have been used.

5.1. Datasets and ranking systems

We used the TREC-8 collections (minus the Congressional Record) for performance evaluation. The dataset contains about 520,000 news documents on 50 topics (No. 401–450) and relevance judgements for the topics. Table 2 shows the number of relevant judged documents for each topic of the dataset. Word stopping and word stemming with single keyword indexing have been performed. Query terms for each topic's initial search (baseline) have been obtained by parsing the title field of a topic. For the baseline and for the first pass ranking (needed for feedback document selection) the default similarity measures provided by Lucene and Lemur have been used (Foundation, 2011; Ogilvie et al., 2002). Performance has been

Table 2
Topics from TREC dataset with number of judged relevant documents available for each topic.

No.	Topic title	# of Relevant docs
401	Foreign minorities, Germany	300
402	Behavioral genetics	80
403	Osteoporosis	21
404	Ireland, peace talks	142
405	Cosmic events	38
406	Parkinson's disease	13
407	Poaching, wildlife preserves	68
408	Tropical storms	118
409	Legal, Pan Am, 103	22
410	Schengen agreement	65
411	Salvaging, shipwreck, treasure	27
412	Airport security	123
413	Steel production	69
414	Cuba, sugar, exports	39
415	Drugs, Golden Triangle	136
416	Three Gorges Project	42
417	Creativity	75
418	Quilts, income	116
419	Recycle, automobile tires	19
420	Carbon monoxide poisoning	33
421	Industrial waste disposal	83
422	Art, stolen, forged	152
423	Milosevic, Mirjana Markovic	21
424	Suicides	171
425	Counterfeiting money	162
426	Law enforcement, dogs	202
427	UV damage, eyes	50
428	Declining birth rates	118
429	Legionnaires' disease	11
430	Killer bee attacks	6
431	Robotic technology	130
432	Profiling, motorists, police	28
433	Greek, philosophy, stoicism	13
434	Estonia, economy	347
435	Curbing population growth	117
436	Railway accidents	180
437	Deregulation, gas, electric	72
438	Tourism, increase	173
439	Inventions, scientific discoveries	219
440	Child labor	54
441	Lyme disease	17
442	Heroic acts	94
443	U.S., investment, Africa	102
444	Supercritical fluids	17
445	Women clergy	62
446	Tourists, violence	162
447	Stirling engine	16
448	Ship losses	46
449	Antibiotics ineffectiveness	67
450	King Hussein, peace	293

measured with TREC's suggested evaluation measures: precision at different levels of retrieved results (P5,10...1000), mean average precision (MAP), R-precision and binary preference (BPREF) (Christopher et al., 2008).

5.2. Explicit and pseudo-relevance feedback setup

In the case of explicit feedback, we take the first M relevant documents from the result set returned by the system after the initial query. Documents are considered relevant or not relevant according to TREC dataset annotations. In contrast, in the case of pseudo-relevance feedback, we take the top M documents retrieved by the system in response to the initial query.

5.3. Parameter tuning

The most important parameters involved in the computation of a WWP structure are the *number of roots* H, the *number of pairs* $|\mathcal{G}|$ and the number of relevant documents M.

Table 3
The number of roots H can be chosen as a trade off between retrieval performances and computation time. 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

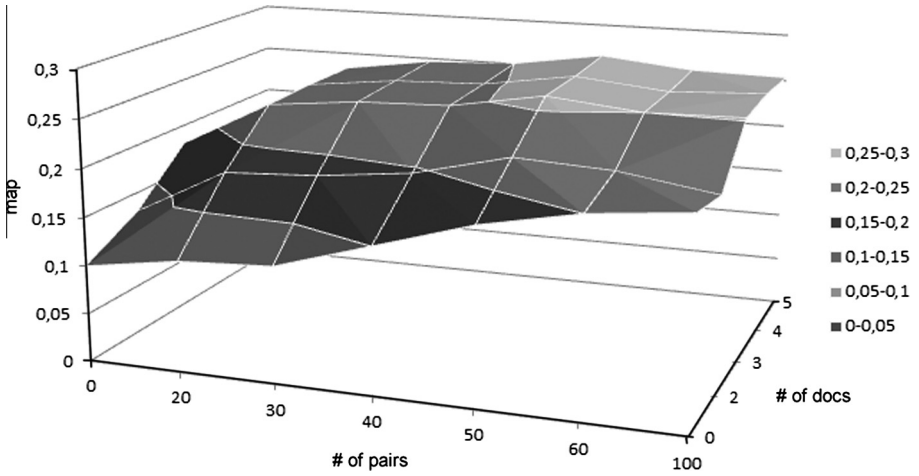

Fig. 5. *WWP* map performance achieved by Lucene by varying the number of pairs and number of relevant documents at the same time.

We have chosen the number of roots $H = 4$ as a trade off between retrieval performances and computation time[1] (see Table 3).

To choose the number M of relevant documents and the number $|\mathcal{G}|$ of pairs, we evaluated the mean average precision (map) achieved by varying at the same time M and $|\mathcal{G}|$. Fig. 5 reveals the relationship between those parameters. In particular, given the number of relevant documents, the performance of the proposed method rapidly increases as number of pairs or increases. The performance increases even when the number of pairs is fixed and number of documents increases, but with a slower speed then the previous case.

However, the highest map values (lightest gray) are obtained only when both, number of documents and pairs, increase. On the contrary, the lowest map value is obtained when both number of documents and pairs are set to zero, that is at the origin of Fig. 5. The origin corresponds to the case of unexpanded query. For the experimentation, we have chosen the values that obtained highest performance, that is: $M = 3$ and $|\mathcal{G}| = 50$.

Note that, the map values showed in Fig. 5 have been obtained with Lucene, but a similar behavior can be observed for Lemur.

5.4. Comparisons with other methods and schemes

5.4.1. Performance analysis with explicit relevance feedback scheme

Table 4 shows the comparison between the performance achieved by the WWP method and the random WWP. The table reports results obtained with both the IR modules, Lucene and Lemur. The random WWP has been obtained by skipping the step 1 of the WWP building process. The probabilities ϕ_{ij} and π_i are randomly set instead of being computed by using the LDA. This test allows to evaluate the reliability of the relations computations with respect to a completely random choice. As can be noted from Table 4, WWP global performance is higher (more than 50%) than the random graph one.

In Table 5 we show results obtained comparing the WWP method with a *Kullback–Leibler* divergence (KLD) based query expansion method (Carpineto et al., 2001) and the baseline. The table reports results obtained with both the IR modules, Lucene and Lemur. As we can see, WWP outperforms KLD and baseline especially for low level of precisions. The improvement of performance is more evident when using Lucene instead of Lemur. These results have obtained without removing feedback documents from the dataset, which is a common behavior for text retrieval systems. However, one could argue that

[1] Results have been obtained using an *Intel Core 2 Duo* 2.40 GHz PC with 4 GB *RAM* with no other process running.

Table 4
Results comparison for *WWP* against random WWP with 3 relevant documents.

IR	Lucene		Lemur	
FE	WWP	WWP (rand)	WWP	WWP (rand)
relret	3068	1472	3285	1577
map	0.2909	0.1236	0.3069	0.1319
Rprec	0.3265	0.1625	0.3324	0.1628
bpref	0.3099	0.1985	0.3105	0.1987
P@5	0.76	0.564	0.736	0.5446
P@10	0.602	0.404	0.58	0.3886
P@100	0.2612	0.1202	0.2562	0.1178
P@1000	0.0614	0.0294	0.0657	0.0315

Table 5
Results comparison for unexpanded query, KLD and WWP (FE) using Lucene and Lemur as IR modules.

IR	Lucene			Lemur		
FE	–	KLD	WWP	–	KLD	WWP
relret	2267	2304	3068	2780	2820	3285
map	0.1856	0.1909	0.2909	0.2447	0.2560	0.3069
Rprec	0.2429	0.2210	0.3265	0.2892	0.2939	0.3324
bpref	0.2128	0.2078	0.3099	0.2512	0.2566	0.3105
P@5	0.3920	0.5200	0.7600	0.4760	0.5720	0.7360
P@10	0.4000	0.4300	0.6020	0.4580	0.4820	0.5800
P@100	0.1900	0.1744	0.2612	0.2166	0.2256	0.2562
P@1000	0.0453	0.0461	0.0614	0.0556	0.0564	0.0657

Table 6
Results comparison for unexpanded query, KLD and WWP using Lucene or Lemur with RSD.

IR	Lucene			Lemur		
FE	–	KLD	WWP	–	KLD	WWP
relret	2117	2178	2921	2630	2668	3143
map	0.1241	0.1423	0.2013	0.1861	0.1914	0.2268
Rprec	0.1862	0.1850	0.2665	0.2442	0.2454	0.2825
bpref	0.1546	0.1716	0.2404	0.1997	0.2044	0.2471
P@5	0.2360	0.3920	0.4840	0.3880	0.4120	0.5120
P@10	0.2580	0.3520	0.4380	0.3840	0.3800	0.4560
P@100	0.1652	0.1590	0.2370	0.1966	0.2056	0.2346
P@1000	0.0423	0.0436	0.0584	0.0526	0.0534	0.0629

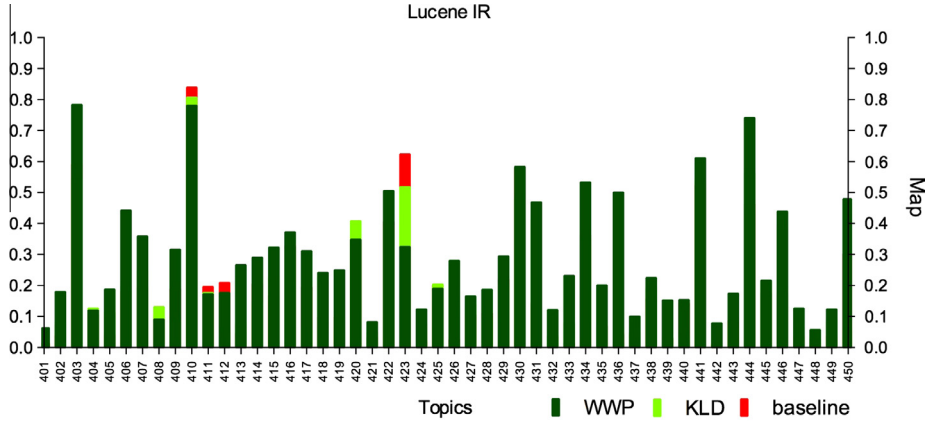

Fig. 6. MAP analysis of WWP, KLD, and baseline for each topic with Lucene IR.

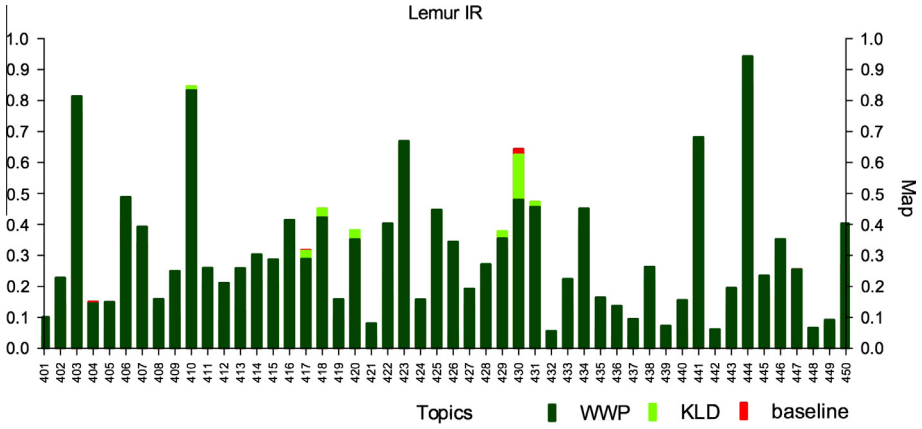

Fig. 7. MAP analysis of WWP, KLD, and baseline for each topic with Lemur.

a big improvement in low level precision is essentially due to the feedback documents that have been better ranked thanks to the use of the query expansion. Therefore, another performance evaluation has been carried out using only the residual collection (RSD), where the feedback documents have been removed. Results for this evaluation are shown in Table 6. It can be noticed that the WWP method achieves better global performance even in the case of residual collection with respect to the other methods.

An average precision analysis for each topic when using Lucene and Lemur IR has been also reported in this work (Figs. 6 and 7). The bar charts allow higher values to hide lower ones so that we can easily identify cases where WWP performs worse than KLD and/or baseline. For example, Fig. 6 shows that the use of an unexpanded query on Lucene for topic 423 achieves better average precision performance than both KLD and WWP. In this case, the use of expanded terms seems to introduce noise in the retrieval task. A similar consideration can be done for Lemur. This analysis demonstrates that the proposed method outperforms other methods in most of the considered query/topics independently of the IR system.

Note that, when Lemur is used (Fig. 7), WWP is able to achieve the best performance in terms of average precision for topic 423 but here we find some issues for topic 430. We observe in some cases that the use of query expansion can have the drawback of reducing performances. Such a behavior can be due to different factors, so that some considerations need to be made. First of all, if we check the number of judged relevant documents available for each topic (Fig. 2), we realize that, for certain topics, such a number is very small compared to the size of the whole dataset. Moreover, not every document in the considered dataset has been judged and there is a large subset of negative examples (document judged as non-relevant) which our system does not take into account to build the graph.

5.4.2. Performance analysis with pseudo relevance feedback

In Table 7 we show the performance obtained by WWP, KLD and baseline in the case of pseudo relevance feedback. As can be noticed, WWP outperforms the other methods but achieves worse performance than the case of explicit feedback. In Table 8 we show the results obtained on the residual collection.

It is well known that the effectiveness of a pseudo relevance approach strictly depends on the quality of retrieved results in response to the initial query. Whatever is the term selection method, WWP or KLD, the relevance of the first *M* documents of the result set can highly compromise the performance of the system. Lucene and Lemur have low values of *precision@10* for several topics that negatively influence performance in several topics as confirmed by the per-query analysis shown in Figs. 8 and 9.

Table 7
Results comparison for unexpanded query, KLD and WWP using Lucene or Lemur. Pseudo-relevance feedback.

IR	Lucene			Lemur		
FE	–	KLD	WWP	–	KLD	WWP
relret	2267	2192	2406	2780	2805	2880
map	0.1856	0.1828	0.1935	0.2447	0.2455	0.2555
Rprec	0.2429	0.2192	0.2449	0.2892	0.2944	0.2962
bpref	0.2128	0.2069	0.2241	0.2512	0.2560	0.2680
P@5	0.3920	0.4320	0.4360	0.4760	0.4880	0.5000
P@10	0.4000	0.3840	0.3920	0.4580	0.4500	0.4620
P@100	0.1900	0.1672	0.1872	0.2166	0.2252	0.2202
P@1000	0.0453	0.0438	0.0481	0.0556	0.0561	0.0576

Table 8
Results comparison for unexpanded query, KLD and WWP using Lucene or Lemur with Residual Set. Pseudo-relevance feedback.

IR	Lucene			Lemur		
FE	–	KLD	WWP	–	KLD	WWP
relret	2117	2137	2346	2630	2741	2815
map	0.1241	0.1628	0.1764	0.1861	0.2255	0.2312
Rprec	0.1862	0.2046	0.2247	0.2442	0.2794	0.2822
bpref	0.1546	0.1912	0.2059	0.1997	0.2373	0.2473
P@5	0.2360	0.4040	0.4160	0.3880	0.4840	0.4840
P@10	0.2580	0.3620	0.3580	0.3840	0.4220	0.4520
P@100	0.1652	0.1596	0.1784	0.1966	0.2170	0.2116
P@1000	0.0423	0.0427	0.0469	0.0526	0.0548	0.0563

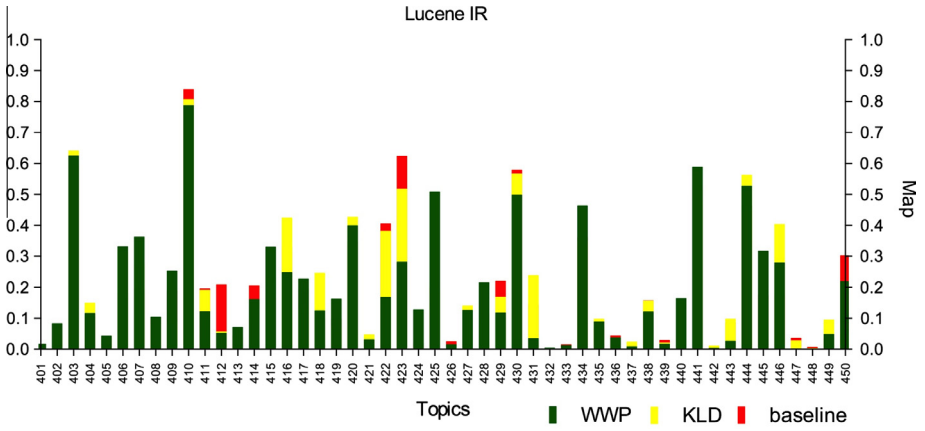

Fig. 8. MAP analysis of WWP, KLD, and baseline for each topic with Lucene IR. Pseudo relevance feedback.

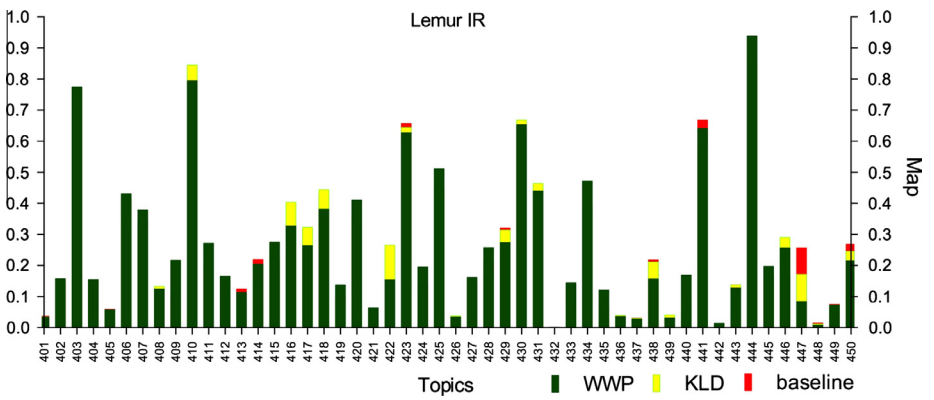

Fig. 9. MAP analysis of WWP, KLD, and baseline for each topic with Lemur. Pseudo relevance feedback.

6. Conclusions

In this work we have demonstrated that a Weighted Word Pairs hierarchical representation is capable of retrieving a greater number of relevant documents than a less complex representation based on a list of words. These results suggest that our approach can be employed in all those text mining tasks that consider matching between patterns represented as textual information and in text categorization tasks as well as in sentiment analysis and detection tasks. The proposed approach computes the expanded queries considering only endogenous knowledge. It is well known that the use of external knowledge, for instance Word- Net, could clearly improve the accuracy of information retrieval systems and we consider this integration as a future work.

Appendix A. Optimization stage

Given the maximum number of roots H and the maximum number of pairs $|\mathcal{G}|$, several WWP structure $\mathbf{g}_t$ 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 $\mathbf{g}_t = \{b_{1t}, \ldots, b_{|\mathcal{G}|t}\}$ in the space $\mathcal{G}$ of the words pairs. Each document of the set Ω_{fback} can be represented as a vector $\mathbf{d}_m = (w_{1m}, \ldots, w_{|\mathcal{G}|m})$ in the space $\mathcal{G}$. A possible scoring function is the cosine similarity between these two vectors:

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

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. Since a score for each document $\mathbf{d}_m$ is obtained, we have:

$$\mathbf{S}_t = \{\mathcal{S}(\mathbf{g}_t, \mathbf{d}_1), \ldots, \mathcal{S}(\mathbf{g}_t, \mathbf{d}_{|\Omega_{fback}|})\},$$

where each score depends on the specific set $\Lambda_t = (\lambda, \mu)_t$. To find the best Λ_t we can maximize the score value for each document, which means that we are looking for the graph which best describes each document of the repository from which it has been extracted. This optimization procedure need 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* ($\mathcal{F}$) will be:

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

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:

$$\Lambda^* = \arg\max_t \{\mathcal{F}(\Lambda_t)\}. \tag{A.2}$$

As discussed before, the space of possible solutions could grow exponentially. For this reason, we considered $|\mathcal{G}| \leq 50$. Moreover, since the number of possible values of Λ_t is in theory infinite, we clustered each set of λ 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.

References

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

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

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, NY, USA: ACM. http://dx.doi.org/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.

Bhogal, J., Macfarlane, A., & Smith, P. (2007). A review of ontology based query expansion. *Information Processing & Management, 43*(4), 866–886. http://dx.doi.org/10.1016/j.ipm.2006.09.003.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 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). 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, NY, USA: ACM. http://dx.doi.org/10.1145/1076034.1076086.

Cao, G., Nie, J.-Y., Gao, J., & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '08* (pp. 243–250). New York, NY, USA: ACM.

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

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys, 44*(1), 1:1–1:50. http://dx.doi.org/10.1145/2071389.2071390.

Christopher, P. R., Manning, D., & Schtze, H. (2008). *Introduction to information retrieval*. Cambridge University.

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: Springer-Verlag.

Colace, F., De Santo, M., Greco, L., & Napoletano, P. (2013). A query expansion method based on a weighted word pairs approach. In *Proceedings of the 3nd Italian Information Retrieval (IIR)* (Vol. 964).

Colace, F., Santo, M. D., Greco, L., & Napoletano, P. (2014). Text classification using a few labeled examples. *Computers in Human Behavior, 30*, 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, NY, USA: ACM.

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.

Efthimiadis, E. N. (1996). Query expansion. In M. E. Williams (Ed.), *Annual review of information systems and technology* (pp. 121–187).

Foundation, A. S. (2011). Apache lucene – scoring, letzter Zugriff: 20. Oktober 2011.

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

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.

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.

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

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, NY, USA: ACM.

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, NY, USA: ACM. http://dx.doi.org/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.), *Information retrieval technology. Lecture notes in computer science* (Vol. 3689, pp. 274–287). Berlin Heidelberg: Springer.

Ogilvie, P., & Callan, J. (2002). Experiments using the lemur toolkit. In *Proceedings of the tenth text retrieval conference* (*TREC-10* (pp. 103–108).

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

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

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

Zhang, J., Deng, B., & Li, X. 2009. Concept based query expansion using wordnet. In *International e-conference on advanced science and technology, 2009. AST 09* (pp. 52–55). http://dx.doi.org/10.1109/AST.2009.24.