A Query Expansion Method based on a Weighted Word Pairs Approach

Francesco Colace[1], Massimo De Santo[1], Luca Greco[1] and Paolo Napoletano[2]

[1] DIEM,University of Salerno, Fisciano,Italy,
desanto@unisa, fcolace@unisa.it, lgreco@unisa.it
[2] DISCo,University of Milano-Bicocca, Italy
napoletano@disco.unimib.it

Abstract. In this paper we propose a query expansion method to improve accuracy of a text retrieval system. Our technique makes use of explicit relevance feedback to expand an initial query with a structured representation called Weighted Word Pairs. Such a structure can be automatically extracted from a set of documents and uses a method for term extraction based on the probabilistic Topic Model. Evaluation has been conducted on TREC-8 repository and performances obtained using standard WWP and Kullback Leibler Divergency query expansion approaches have been compared.

Keywords: Text retrieval, query expansion, probabilistic topic model

1 Introduction

Over the years, several text retrieval models have been proposed: set-theoretic (including boolean), algebraic, probabilistic models [1], etc. Although each method has its own properties, there is a common denominator: the *bag of words* representation of documents.

The "bag of words" assumption claims that a document 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 [1]. The elements of the vector can be weights and computed in different ways so that a document can be considered as a list of weighted features. The *term frequency-inverse document (tf-idf)* model is a commonly used weighting model: each term in a document collection is weighted by measuring how often it is found within a document (*term frequency*), offset by how often it occurs within the entire collection (*inverse document frequency*). Based on this model, also a query can be viewed as a document, so it can be represented as a vector of weighted words.

The relevance of a document to a query is the distance between the corresponding vector representations in the features space. Unfortunately, queries performed by users may not be long enough to avoid the inherent ambiguity of language (polysemy etc.). This makes text retrieval systems, that rely on the

bags of words model, generally suffer from low precision, or low quality document retrieval. To overcome this problem, scientists proposed methods to expand the original query with other topic-related terms extracted from *exogenous* (e.g. ontology, WordNet, data mining) or *endogenous* knowledge (i.e. extracted only from the documents contained in the collection) [2, 3, 1]. Methods based on *endogenous* knowledge, also known as relevance feedback, make use of a number of labelled documents, provided by humans (explicit) or automatic/semi-automatic strategies, to extract topic-related terms and such methods have demonstrated to obtain performance improvements of up to 40% [4]

In this paper we propose a new query expansion method that uses a structured representation of documents and queries, named *Weighted Word Pairs*, that is capable of reducing the effect of the inherent ambiguity of language so achieving better performance than a method based on a vector of weighted words. The *Weighted Word Pairs* representation is automatically obtained from documents, provided by a minimal explicit feedback, by using a method of *term extraction*[5][6][7] based on the *Latent Dirichlet Allocation* model [8] implemented as the *Probabilistic Topic Model* [9]. Evaluation has been conducted on TREC-8 repository: results obtained employing standard WWP and Kullback Leibler divergency have been compared.

This article is structured as follows: Section 2 gives an overview on related works and approaches to query expansion in text retrieval; in Section 3 a general framework for query expansion is discussed; Section 4 describes in detail our feature extraction method; in Section 5 performance evaluation is presented.

2 Related works

It is well documented that the query length in typical information retrieval systems is rather short (usually two or three words) [10] which may not be long enough to avoid the inherent ambiguity of language (polysemy etc.), and which makes text retrieval systems, that rely on a term-frequency based index, suffer generally from low precision, or low quality of document retrieval.

In turn, the idea of taking advantage of additional knowledge, by expanding the original query with other topic-related terms, to retrieve relevant documents has been largely discussed in the literature, where manual, interactive and automatic techniques have been proposed [2][1]. The idea behind these techniques is that, in order to avoid ambiguity, it may be sufficient to better specify "the meaning" of what the user has in mind when performing a search, or in other words "the main concept" (or a set of concepts) of the preferred topic in which the user is interested. A better specialization of the query can be obtained with additional knowledge, that can be extracted from *exogenous* (e.g. ontology, WordNet, data mining) or *endogenous* knowledge (i.e. extracted only from the documents contained in the repository) [3, 1].

In this paper we focus on those techniques which make use of the *Relevance Feedback* (in the case of endogenous knowledge) which takes into account the results that are initially returned from a given query and so uses the information

about the relevance of each result to perform a new expanded query. In the literature we can distinguish between three types of procedures for the assignment of the relevance: explicit feedback, implicit feedback, and pseudo feedback.

Most existing methods, due to the fact that the human labeling task is enormously annoying and time consuming [11], make use of the pseudo relevance feedback (top k retrieved are assumed to be relevant). Nevertheless, fully automatic methods suffer from obvious errors when the initial query is intrinsically ambiguous. As a consequence, in the recent years, some hybrid techniques have been developed which take into account a minimal explicit human feedback [4, 12] and use it to automatically identify other topic related documents.

However, 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 [13]. 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 [14], [15], [16].

3 A general Query Expansion framework

A general query expansion framework can be described as a modular system including:

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

Such a framework is represented in Figure 1 and can be described as follows. The user initially performs a search task on the dataset $\mathcal{D}$ by inputting a query $\mathbf{q}$ to the IR system and obtains a set of documents $\mathcal{RS} = (\mathbf{d}_1, \cdots, \mathbf{d}_N)$ as a result. The module F, thanks to the explicit feedback of the user, identifies a small set of relevant documents (called *Relevance Feedback*) $\mathcal{RF} = (\mathbf{d}_1, \cdots, \mathbf{d}_M)$ from the hit list of documents $\mathcal{RS}$ returned by the IR system. Given the set of relevant document $\mathcal{RF}$, the module FE extracts a set of features $\mathbf{g}$ that must be added to the initial query $\mathbf{q}$. The extracted features can be weighted words or more complex structures such as weighted word pairs. So the obtained set $\mathbf{g}$ must be adapted by the QR module to be handled by the IR system and then added to the initial query. The output of this module is a new query $\mathbf{qe}$ which includes both the initial query and the set of features extracted from the $\mathcal{RF}$. The new query is then performed on the collection so obtaining a new result set $\mathcal{RS}' = (\mathbf{d'}_1, \cdots, \mathbf{d'}_N)$ different from the one obtained before. Considering the framework described above is possible to take into account any technique of feature extraction that makes use of the explicit relevant feedback and any IR

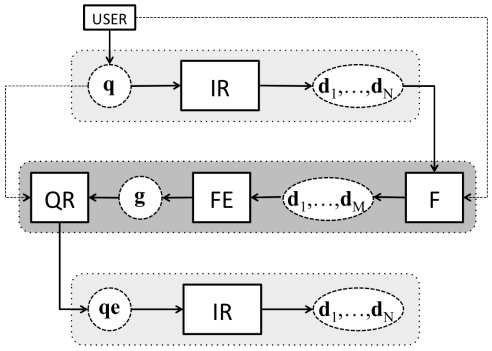

Fig. 1. General framework for Query Expansion.

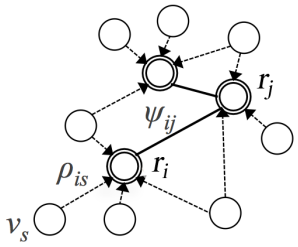

Fig. 2. Graphical representation of a *Weighted Word Pairs* structure.

systems suitable to handle the resulting expanded query **qe**. In this way it is possible to implement several techniques and make objective comparisons with the proposed one.

4 WWP feature selection method

The aim of the proposed method is to extract from a set of documents a compact representation, named *Weighted Word Pairs* (WWP), which contains the most discriminative word pairs to be used in the text retrieval task. The Feature Extraction module (FE) is represented in Fig. 3. The input of the system is the set of documents $\mathcal{RF} = (\mathbf{d}_1, \cdots, \mathbf{d}_M)$ and the output is a vector of weighted word pairs $\mathbf{g} = \{w'_1, \cdots, w'_{|\mathcal{T}_p|}\}$, where $\mathcal{T}_p$ is the number of pairs and w'_n is the weight associated to each pair (feature) $t_n = (v_i, v_j)$.

A WWP structure can be suitably represented as a *graph* **g** of terms (Fig. 2). Such a graph is made of several clusters, each containing a set of words v_s (*aggregates*) related to an *aggregate root* (r_i), a special word which represents the centroid of the cluster. How aggregate roots are selected will be clear further. The weight ρ_{is} can measure how a word is related to an aggregate root and can be expressed as a probability: $\rho_{is} = P(r_i|v_s)$. The resulting structure is a subgraph rooted on r_i. Moreover, *aggregate roots* can be linked together

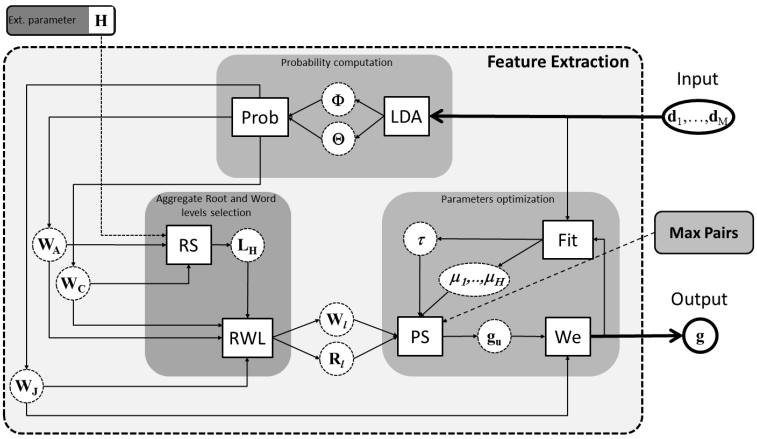

Fig. 3. Proposed feature extraction method. A *Weighted Word Pairs* **g** structure is extracted from a corpus of training documents.

building a centroids subgraph. The weight ψ_{ij} can be considered as the degree of correlation between two aggregate roots and can also be expressed as a probability: $\psi_{ij} = P(r_i, r_j)$. Being each aggregate root a special word, it can be stated that **g** contains directed and undirected pairs of features lexically denoted as words. Given the training set $\mathcal{RF}$ of documents, the term extraction procedure is obtained first by computing all the relationships between words and aggregate roots (ρ_{is} and ψ_{ij}), and then selecting the right subset of pairs $\mathcal{T}_{sp}$ from all the possible ones $\mathcal{T}_p$.

A WWP graph **g** is learned from a corpus of documents as a result of two important phases: the *Relations Learning* stage, where graph relation weights are learned by computing probabilities between word pairs (see Fig. 3); the *Structure Learning* stage, where an initial WWP graph, which contains all possible relations between aggregate roots and aggregates, is optimized by performing an iterative procedure. Given the number of aggregate roots H and the desired max number of pairs as constraints, the algorithm chooses the best parameter settings $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_H)$ and τ defined as follows:

1. μ_i: the threshold that establishes, for each aggregate root i, the number of *aggregate root/word* pairs of the graph. A relationship between the word v_s and the aggregate root r_i is relevant if $\rho_{is} \geq \mu_i$.
2. τ: the threshold that establishes the number of *aggregate root/aggregate root* pairs of the graph. A relationship between the aggregate root v_i and aggregate root r_j is relevant if $\psi_{ij} \geq \tau$.

4.1 Relations Learning

Since each aggregate root is lexically represented by a word of the vocabulary, we can write $\rho_{is} = P(r_i|v_s) = P(v_i|v_s)$, and $\psi_{ij} = P(r_i, r_j) = P(v_i, v_j)$. Considering that $P(v_i, v_j) = P(v_i|v_j)P(v_j)$, all the relations between words result from the computation of the joint or the conditional probability $\forall i, j \in \{1, \cdots, |\mathcal{T}|\}$ and $P(v_j)$ $\forall j$. An exact calculation of $P(v_j)$ and an approximation of the joint, or conditional, probability can be obtained through a smoothed version of the generative model introduced in [8] called Latent Dirichlet Allocation (LDA), which makes use of Gibbs sampling [9]. The original theory introduced in [9] mainly proposes a semantic representation in which documents are represented in terms of a set of probabilistic topics z. Formally, we consider a word u_m of the document $\mathbf{d}_m$ as a random variable on the vocabulary $\mathcal{T}$ and z as a random variable representing a topic between $\{1, \cdots, K\}$. A document $\mathbf{d}_m$ results from generating each of its words. 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 $\mathcal{RF}$ consists of two matrixes:

1. the *words-topics* matrix 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 that contains $K \times |\mathcal{RF}|$ 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 within a document $\mathbf{d}_m$ of the corpus can be then obtained as:

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

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:

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

Note that the exact calculation of Eq. 2 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. 2 can be written as [5, 6]:

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

Moreover, Eq. 1 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:

$$P(u) = \sum_{m=1}^{M} P(u_m)\delta_m \tag{4}$$

where δ_m is the prior probability for each document ($\sum_{m=1}^{|\mathcal{RF}|} \delta_m = 1$). If we consider the joint probability distribution of two words u and y, we obtain:

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

Concluding, once we have $P(u)$ and $P(u, y)$ we can compute $P(v_i) = P(u = v_i)$ and $P(v_i, v_j) = P(u = v_i, y = v_j)$, $\forall i, j \in \{1, \cdots, |\mathcal{T}|\}$ and so the relations learning can be totally accomplished.

4.2 Structure Learning

Once each ψ_{ij} and ρ_{is} is known $\forall i, j, s$, aggregate root and word levels have to be identified in order to build a starting WWP structure to be optimized as discussed later. The first step is to select from the words of the indexed corpus a set of aggregate roots $\mathbf{r} = (r_1, \ldots, r_H)$, which will be the nodes of the centroids subgraph. Aggregate roots are meant to be the words whose occurrence is most implied by the occurrence of other words of the corpus, so they can be chosen as follows:

$$r_i = \text{argmax}_{v_i} \prod_{j \neq i} P(v_i|v_j) \tag{6}$$

Since relationships' strenghts between aggregate roots can be directly obtained from ψ_{ij}, the centroids subgraph can be easily determined. Note that not all possible relationships between aggregate roots are relevant: the threshold τ can be used as a free parameter for optimization purposes. As discussed before, several words (aggregates) can be related to each aggregate root, obtaining H aggregates' subgraphs. The threshold set $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_H)$ can be used to select the number of relevant pairs for each aggregates' subgraph. Note that a relationship between the word v_s and the aggregate root r_i is relevant if $\rho_{is} \geq \mu_i$, but the value ρ_{is} cannot be directly used to express relationships' strenghts between aggregate roots and words. In fact, being ρ_{is} a conditional probability, it is always bigger than ψ_{is} which is a joint probability. Therefore, once pairs for the aggregates' subgraph are selected using ρ_{is}, relationships' strenght are represented on the WWP structure through ψ_{is}.
Given H and the maximum number of pairs as constraints (i.e. fixed by the user), several WWP structure $\mathbf{g}_t$ can be obtained by varying the parameters $\Lambda_t = (\tau, \mu)_t$. As shown in Fig.3, an optimization phase is carried out in order to search the set of parameters Λ_t which produces the best WWP graph

[6]. This process relies on a scoring function and a searching strategy that will be now explained. As we have previously seen, a $\mathbf{g}_t$ is a vector of features $\mathbf{g}_t = \{b_{1t}, \ldots, b_{|\mathcal{T}_{sp}|t}\}$ in the space $\mathcal{T}_{sp}$ and each document of the training set $\mathcal{RF}$ can be represented as a vector $\mathbf{d}_m = (w_{1m}, \ldots, w_{|\mathcal{T}_{sp}|m})$ in the space $\mathcal{T}_{sp}$. A possible scoring function is the cosine similarity between these two vectors:

$$\mathcal{S}(\mathbf{g}_t, \mathbf{d}_m) = \frac{\sum_{n=1}^{|\mathcal{T}_{sp}|} b_{nt} \cdot w_{nm}}{\sqrt{\sum_{n=1}^{|\mathcal{T}_{sp}|} b_{nt}^2} \cdot \sqrt{\sum_{n=1}^{|\mathcal{T}_{sp}|} w_{nm}^2}} \tag{7}$$

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 $\mathcal{RF}$ is the one that produces the maximum score attainable for each document when used to rank $\mathcal{RF}$ documents. Since a score for each document $\mathbf{d}_m$ is obtained, we have:

$$\mathbf{S}_t = \{\mathcal{S}(\mathbf{g}_t, \mathbf{d}_1), \cdots, \mathcal{S}(\mathbf{g}_t, \mathbf{d}_{|\mathcal{RF}|})\},$$

where each score depends on the specific set $\Lambda_t = (\tau, \mu)_t$. To compute the best value of Λ 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 learned. It should be noted that such an optimization maximizes at the same time all $|\mathcal{RF}|$ elements of $\mathbf{S}_t$. 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 two-objective one. Additionally, the latter problem can be reformulated by means of a linear combination of its objectives, thus obtaining a single objective function, i.e., *Fitness* ($\mathcal{F}$), which depends on Λ_t,

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

where E is the mean value of all the elements of $\mathbf{S}_t$ and σ_m is the standard deviation. By summing up, the parameters learning procedure is represented as follows, $\Lambda^* = \text{argmax}_t\{\mathcal{F}(\Lambda_t)\}$.

Since the space of possible solutions could grow exponentially, $|\mathcal{T}_{sp}| \leq 300$ [3] has been considered. Furthermore, the remaining space of possible solutions has been reduced by applying a clustering method, that is the *K-means* algorithm, to all ψ_{ij} and ρ_{is} values, so that the optimum solution can be exactly obtained after the exploration of the entire space.

5 Method validation

The proposed approach has been validated using IR systems that allow to handle structured queries composed of weighted word pairs. For this reason,the following open source tools were considered: Apache Lucene[4] which supports structured

[3] This number is usually employed in the case of Support Vector Machines.

[4] We adopted the version 2.4.0 of Lucene

query based on a weighted boolean model and Indri[5] which supports an extended set of probabilistic structured query operators based on INQUERY. The performance comparison was carried out testing the following FE/IR configurations:

- **IR only**. Unexpanded queries were 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 was used to expand initial query and feed Lucene and Lemur IR modules.
- **FE(KLD) + IR**. Kullback Leibler Divergency based feature extraction was used to expand initial query and feed Lucene and Lemur IR modules.

5.1 Datasets and Ranking Systems

The dataset from TREC-8 [17] collections (minus the Congressional Record) was used for performance evaluation. It contains about 520,000 news documents on 50 topics (no.401-450) and relevance judgements for the topics. Word stopping and word stemming with single keyword indexing were performed. Query terms for each topic's initial search (baseline) were 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 has been used. Performance was measured with TREC's standard evaluation measures: mean average precision (MAP), precision at different levels of retrieved results (P@5,10...1000), R-precision and binary preference (BPREF).

5.2 Parameter Tuning

The two most important parameters involved in the computation of WWP, given the number of documents for training, are the *number of aggregate roots H* and the *number of pairs*. The number of aggregate roots can be chosen as a trade off between retrieval performances and computational times, our choice was $H = 4$ since it seemed to be the best compromise (about 6 seconds per topic)[6]. However, we want to emphasize method effectiveness more than algorithm efficiency since algorithm coding has not been completely optimized yet.

Fig. 5.2 shows results of baseline and WWP method when changing *number of pairs* from 20 to 100 where the number of documents is fixed to 3: in this analysis, Lucene IR module is used . According to the graph, our system always provides better performances than baseline; the change in number of pairs has a great impact especially on precision at 5 where 60 pairs achieve the best results. Anyway, if we consider precision at higher levels together with map values, 50 pairs seem to be a better choice also for shorter computational times. Fig. 5.2 shows results of baseline and our method when changing *number of training documents* (Lucene IR Module used): here we can see that the overall behaviour of the system is better when choosing 3 relevant documents for training.

[5] We adopted the version 5... that is part of the Lemur Toolkit

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

Once again the system outperforms baseline especially at low precision levels. Discussed analysis led us to choose the following settings for the experimental stage: 4 aggregate roots, 50 pairs, 3 training documents.

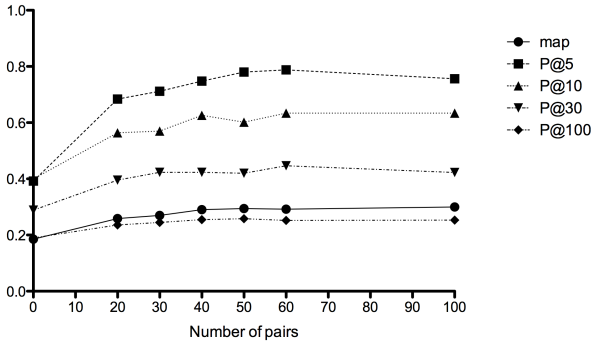

Fig. 4. *WWP* performance when changing number of pairs.

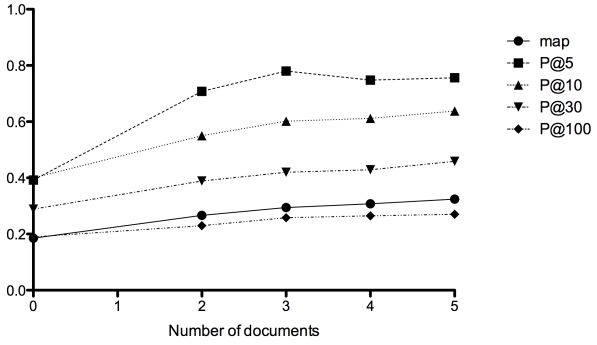

Fig. 5. *WWP* performance when changing number of training documents

5.3 Comparison with other methods

In Table 1 WWP method is compared with baseline and *Kullback-Leibler* divergence based method [13] when using both Lucene and Lemur as IR modules. Here we see that WWP outscores KLD, and baseline especially for low level precision while having good performances for other measures. However these results are obtained without removing feedback documents from the dataset so

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 1. 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	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

Table 2. Results comparison for unexpanded query, KLD and WWP using Lucene or Lemur with RSD.

a big improvement in low level precision may appear a little obvious. Another performance evaluation was carried out using only the residual collection (RSD) where feedback documents are removed. Results for this evaluation are shown in table 2 where we see performance improvements also with residual collection.

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.

References

1. Christopher D. Manning, P.R., Schtze, H.: Introduction to Information Retrieval. Cambridge University (2008)
2. Efthimiadis, E.N.: Query expansion. In Williams, M.E., ed.: Annual Review of Information Systems and Technology. (1996) 121–187
3. Bhogal, J., Macfarlane, A., Smith, P.: A review of ontology based query expansion. Information Processing & Management **43**(4) (2007) 866 – 886
4. Okabe, M., Yamada, S.: Semisupervised query expansion with minimal feedback. IEEE Transactions on Knowledge and Data Engineering **19** (2007) 1585–1589
5. Napoletano, P., Colace, F., De Santo, M., Greco, L.: Text classification using a graph of terms. In: Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on. (july 2012) 1030 –1035
6. Clarizia, F., Greco, L., Napoletano, P.: An adaptive optimisation method for automatic lightweight ontology extraction. In Filipe, J., Cordeiro, J., eds.: Enterprise Information Systems. Volume 73 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2011) 357–371
7. Clarizia, F., Greco, L., Napoletano, P.: A new technique for identification of relevant web pages in informational queries results. In: Proceedings of the 12th International Conference on Enterprise Information Systems: Databases and Information Systems Integration. (8-12 June 2010) 70–79
8. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3**(993–1022) (2003)
9. Griffiths, T.L., Steyvers, M., Tenenbaum, J.B.: Topics in semantic representation. Psychological Review **114**(2) (2007) 211–244
10. Jansen, B.J., Spink, A., Saracevic, T.: Real life, real users, and real needs: a study and analysis of user queries on the web. Inform. Proces. & Manag. **36**(2) (2000) 207–227
11. Ko, Y., Seo, J.: Text classification from unlabeled documents with bootstrapping and feature projection techniques. Inf. Process. Manage. **45** (2009) 70–83
12. Dumais, S., Joachims, T., Bharat, K., Weigend, A.: SIGIR 2003 workshop report: implicit measures of user interests and preferences. **37**(2) (2003) 50–54
13. Carpineto, C., de Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Trans. Inf. Syst. **19** (2001) 1–27
14. Callan, J., Croft, W.B., Harding, S.M.: The inquery retrieval system. In: In Proceedings of the Third International Conference on Database and Expert Systems Applications, Springer-Verlag (1992) 78–83
15. Collins-Thompson, K., Callan, J.: Query expansion using random walk models. In: Proceedings of the 14th ACM international conference on Information and knowledge management. CIKM '05, New York, NY, USA, ACM (2005) 704–711
16. Lang, H., Metzler, D., Wang, B., Li, J.T.: Improved latent concept expansion using hierarchical markov random fields. In: Proceedings of the 19th ACM international conference on Information and knowledge management. CIKM '10, New York, NY, USA, ACM (2010) 249–258
17. Voorhees, E.M., Harman, D.: Overview of the eighth text retrieval conference (trec-8) (2000)