Selecting Good Expansion Terms for Pseudo-Relevance Feedback

Guihong Cao, Jian-Yun Nie
Department of Computer Science and Operations Research
University of Montreal, Canada
{caogui, nie}@iro.umontreal.ca

Jianfeng Gao
Microsoft Research, Redmond, USA
jfgao@microsoft.com

Stephen Robertson
Microsoft Research at Cambridge, Cambridge, UK
ser@microsoft.com

ABSTRACT

Pseudo-relevance feedback assumes that most frequent terms in the pseudo-feedback documents are useful for the retrieval. In this study, we re-examine this assumption and show that it does not hold in reality – many expansion terms identified in traditional approaches are indeed unrelated to the query and harmful to the retrieval. We also show that good expansion terms cannot be distinguished from bad ones merely on their distributions in the feedback documents and in the whole collection. We then propose to integrate a term classification process to predict the usefulness of expansion terms. Multiple additional features can be integrated in this process. Our experiments on three TREC collections show that retrieval effectiveness can be much improved when term classification is used. In addition, we also demonstrate that good terms should be identified directly according to their possible impact on the retrieval effectiveness, i.e. using supervised learning, instead of unsupervised learning.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Retrieval models

General Terms

Design, Algorithm, Theory, Experimentation

Keywords

Pseudo-relevance feedback, Expansion Term Classification, SVM, Language Models

1. INTRODUCTION

User queries are usually too short to describe the information need accurately. Many important terms can be absent from the query, leading to a poor coverage of the relevant documents. To solve this problem, query expansion has been widely used [9], [15], [21], [22]. Among all the approaches, pseudo-relevance feedback (PRF) exploiting the retrieval result has been the most effective [21]. The basic assumption of PRF is that the top-ranked documents in the first retrieval result contain many useful terms that can help discriminate relevant documents from irrelevant ones. In general, the expansion terms are extracted either according to the term distributions in the feedback documents (i.e. one tries to extract the most frequent terms); or according to the comparison between the term distributions in the feedback documents and in the whole document collection (i.e. to extract the most specific terms in the feedback documents). Several additional criteria have been proposed. For example, *idf* is widely used in vector space model [15]. Query length has been considered in [7] for the weighting of expansion terms. Some linguistic features have been tested in [16].

However, few studies have directly examined whether the expansion terms extracted from pseudo-feedback documents by the existing methods can indeed help retrieval. In general, one was concerned only with the global impact of a set of expansion terms on the retrieval effectiveness.

A fundamental question often overlooked at is whether the expansion terms extracted are truly related to the query and are useful for IR. In fact, as we will show in this paper, the assumption that most expansion terms extracted from the feedback documents are useful does not hold, even when the global retrieval effectiveness can be improved. Among the extracted terms, a non-negligible part is either unrelated to the query or is harmful, instead of helpful, to retrieval effectiveness. So a crucial question is: how can we better select useful expansion terms from pseudo-feedback documents?

In this study, we propose to use a supervised learning method for term selection. The term selection problem can be considered as a term classification problem – we try to separate good expansion terms from the others directly according to their potential impact on the retrieval effectiveness. This method is different from the existing ones, which can typically be considered as an unsupervised learning. SVM [6], [20] will be used for term classification, which uses not only the term distribution criteria as in previous studies, but also several additional criteria such as term proximity.

This approach proposed has at least the following advantages: 1) Expansion terms are no longer selected merely based on term distributions and other criteria indirectly related to the retrieval effectiveness. It is done directly according to their possible impact on the retrieval effectiveness. We can expect the selected terms to have a higher impact on the effectiveness. 2) The term classification process can naturally integrate various criteria, and thus provides a framework for incorporating different sources of evidence.

We evaluate our method on three TREC collections and compare it to the traditional approaches. The experimental results show that the retrieval effectiveness can be improved significantly when term classification is integrated. To our knowledge, this is the first attempt trying to investigate the direct impact on retrieval effectiveness of individual expansion terms in pseudo-relevance feedback.

The remaining of the paper is organized as follows: Section 2 reviews some related work and the state-of-the-art approaches to query expansion. In section 3, we examine the PRF assumption used in the previous studies and show that it does not hold in

Conference'08, July 20–24, 2008, Singapore.

reality. Section 4 presents some experiments to investigate the potential usefulness of selecting good terms for expansion. Section 5 describes our term classification method and reports an evaluation of the classification process. The integration of the classification results into the PRF methods is described in Section 6. In section 7, we evaluate the resulting retrieval method with three TREC collections. Section 8 concludes this paper and suggests some avenues for future work.

2. Related Work

Pseudo-relevance feedback has been widely used in IR. It has been implemented in different retrieval models: vector space model [15], probabilistic model [13], and so on. Recently, the PRF principle has also been implemented within the language modeling framework. Since our work is also carried out using language modeling, we will review the related studies in this framework in more detail.

The basic ranking function in language modeling uses KL-divergence as follows:

$$Score(d,q) = \sum_{w \in V} P(w|\theta_q) log P(w|\theta_d) \qquad (1)$$

where V is the vocabulary of the whole collection, θ_q and θ_d are respectively the query model and the document model. The document model has to be smoothed to solve the zero-probability problem. A commonly used smoothing method is Dirichlet smoothing [23]:

$$P(w|\theta_d) = \frac{tf(w,d) + uP(w|C)}{|d| + u} \qquad (2)$$

where $|d|$ is the length of the document, $tf(w,d)$ the term frequency of w within d, $P(w|C)$ the probability of w in the whole collection C estimated with MLE (Maximum Likelihood Estimation), and u is the Dirichlet prior (set at 1,500 in our experiments).

The query model describes the user's information need. In most traditional approaches using language modeling, this model is estimated with MLE without smoothing. We denote this model by$P(w|\theta_o)$. In general, this query model has a poor coverage of the relevant and useful terms, especially for short queries. Many terms related to the query's topic are absent from (or has a zero probability in) the model. Pseudo-relevance feedback is often used to improve the query model. We mention two representative approaches here: relevance model and mixture model.

The relevance model [8] assumes that a query term is generated by a relevance model $P(w|\theta_R)$. However, it is impossible to define the relevance model without any relevance information. [8] thus exploits the top-ranked feedback documents by assuming them to be samples from the relevance model. The relevance model is then estimated as follows:

$$P(w|\theta_R) \approx \sum_{D \in \mathcal{F}} P(w|D) P(D|\theta_R)$$

Where $\mathcal{F}$ denotes the feedback documents. On the right side, the relevance model θ_R is approximated by the original query Q. Applying Bayesian rule and making some simplifications, we obtain:

$$P(w|\theta_R) \approx \frac{\sum_{D \in \mathcal{F}} P(w|D) P(Q|D) P(D)}{P(Q)} \propto \sum_{D \in \mathcal{F}} P(w|D) P(Q|D) \qquad (3)$$

That is, the probability of a term w in the relevance model is determined by its probability in the feedback documents (i.e. $P(w|D)$) as well as the correspondence of the latter to the query (i.e. $P(Q|D)$). The above relevance model is used to enhance the original query model by the following interpolation:

$$P(w|\theta_q) = (1-\lambda) P(w|\theta_o) + \lambda P(w|\theta_R) \qquad (4)$$

where λ is the interpolation weight (set at 0.5 in our experiments). Notice that the above interpolation can also be implemented as document re-ranking in practice, in which only the top-ranked documents are re-ranked according to the relevance model.

The mixture model [22] also tries to build a language model for the query topic from the feedback documents, but in a way different from the relevance model. It assumes that the query topic model $P(w|\theta_T)$ to be extracted corresponds to the part that is the most distinctive from the whole document collection. This distinctive part is extracted as follows: Each feedback document is assumed to be generated by the topic model to be extracted and the collection model, and the EM algorithm [3] is used to extract the topic model so as to maximize the likelihood of the feedback documents. Then the topic model is combined with the original query model by an interpolation similarly to the relevance model.

Although the specific techniques used in the above two approaches are different, both assume that the strong terms contained in the feedback documents are related to the query and are useful to improve the retrieval effectiveness. In both cases, the strong terms are determined according to their distributions. The only difference is that the relevant model tries to extract the most frequent terms from the feedback documents (i.e. with a strong $P(w|D)$), while the mixture model tries to extract those that are the most distinctive between the feedback documents and the general collection. These criteria have been generally used in other PRF approaches (e.g. [21]).

Several additional criteria have been used to select terms related to the query. For example, [14] proposed the principle that the selected terms should have a higher probability in the relevant documents than in the irrelevant documents. For document filtering, term selection is more widely used in order to update the topic profile. For example, [24] extracted terms from true relevant and irrelevant documents to update the user profile (i.e. query) using the Rocchio method. Kwok et al. [7] also made use of the query length as well as the size of the vocabulary. Smeaton and Van Rijsbergen [16] examined the impact of determining expansion terms using minimal spanning tree and some simple linguistic analysis.

Despite the large number of studies, a crucial question that has not been directly examined is whether the expansion terms selected in a way or another are truly useful for the retrieval. One was usually concerned with the global impact of a set of expansion terms. Indeed, in many experiments, improvements in the retrieval effectiveness have been observed with PRF [8], [9], [19], [22]. This might suggest that most expansion terms are useful. Is it really so in reality? We will examine this question in the next section.

Notice that some studies (e.g. [11]) have tried to understand the effect of query expansion. However, these studies have examined the terms extracted from the whole collection instead of from the feedback documents. In addition, they also focused on the term distribution aspects.

3. A Re-examination of the PRF Assumption

The general assumption behind PRF can be formulated as follows:

Most frequent or distinctive terms in pseudo-relevance feedback documents are useful and they can improve the retrieval effectiveness when added into the query.

To test this assumption, we will consider all the terms extracted from the feedback documents using the mixture model. We will test each of these terms in turn to see its impact on the retrieval effectiveness. The following score function is used to integrate an expansion term *e*:

$$Score(d,q) = \sum_{t \in q} P(t|\theta_o) logP(t|\theta_d) + wlogP(e|\theta_d) \qquad (5)$$

where t is a query term, $P(t|\theta_o)$ is the original query model as described in section 2, *e* is the expansion term under consideration, and *w* is its weight. The above expression is a simplified form of query expansion with a single term. In order to make the test simpler, the following simplifications are made: 1) An expansion term is assumed to act on the query independently from other expansion terms; 2) Each expansion term is added into the query with equal weight - the weight *w* is set at 0.01 or -0.01. In practice, an expansion term may act on the query in dependence with other terms, and their weights may be different. Despite these simplifications, our test can still reflect the main characteristics of the expansion terms.

Good expansion terms are those that improve the effectiveness when *w* is 0.01 and hurt the effectiveness when *w* is -0.01; bad expansion terms produce the opposite effect. Neutral expansion terms are those that produce similar effect when *w* is 0.01 or -0.01. Therefore we can generate three groups of expansion terms: good, bad and neutral. Ideally, we would like to use only good expansion terms to expand queries.

Let us describe the identification of the three groups of terms in more detail. Suppose *MAP(q)* and $MAP(q \cup e)$ are respectively the MAP of the original query and expanded query (expanded with *e*). We measure the performance change due to *e* by the ratio $chg(e) = \frac{MAP(q \cup e) - MAP(q)}{MAP(q)}$. We set a threshold at 0.005, i.e., good and bad expansion terms should produce a performance change such that $|chg(e)| > 0.005$.

In addition to the above performance change, we also assume that a term appearing less than 3 times in the feedback documents is not an important expansion term. This allows us to filter out some noise.

The above identification produces a desired result for term classification. Now, we will examine whether the candidate expansion terms proposed by the mixture model are good terms. Our verification is made on three TREC collections: AP, WSJ and Disk4&5. The characteristics of these collections are described in Section 7.1. We consider 150 queries for each collection and 80 expansions with the largest probabilities for each query. The following table shows the proportion of good, bad and neutral terms for all the queries in each collection.

Collection	Good Terms	Neutral Terms	Bad Terms
AP	17.52%	47.59%	36.69%
WSJ	17.41%	49.89%	32.69%
Disk4&5	17.64%	56.46%	25.88%

Table 1. Proportions of each group of expansion terms selected by the mixture model

As we can see, only less than 18% of the expansion terms used in the mixture model are good terms in all the three collections. The proportion of bad terms is higher. This shows that the expansion process indeed added more bad terms than good ones.

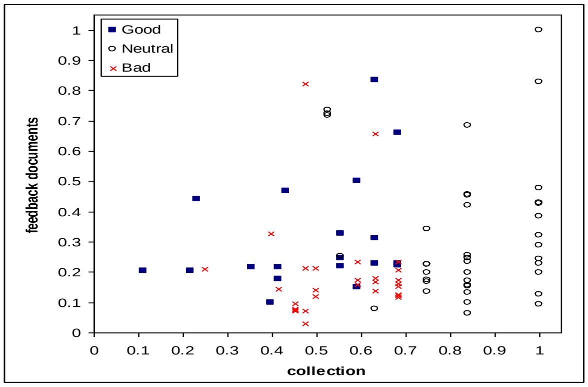

Figure 1. Distribution of the expansion terms for "airbus subsidies" in the feedback documents and in the collection

We also notice from Table 1 that a large proportion of the expansion terms are neutral terms, which have little impact on the retrieval effectiveness. Although this part of the terms does necessarily not hurt retrieval, adding them into the query would produce a long query and thus a heavier query traffic (longer evaluation time). It is then desirable to remove these terms, too.

The above analysis clearly shows that the term selection process used in the mixture model is insufficient. Similar phenomenon is observed on the relevance model and can be generalized to all the methods exploiting the same criteria. This suggests that the term selection criteria used - term distributions in the feedback documents and in the whole document collection, is insufficient. This also indicates that good and bad expansion terms may have similar distributions because the mixture model, which exploits the difference of term distribution between the feedback documents and the collection, has failed to distinguish them.

To illustrate the last point, let us look at the distribution of the expansion terms selected with the mixture model for TREC query #51 "airbus subsidies". In Figure 1, we place the top 80 expansion terms with the largest probabilities in a two-dimensional space – one dimension represents the logarithm of its probability in the pseudo-relevant documents and another dimension represents that in the whole collection. To make the illustration easier, a simple normalization is made so that the final value will be in the range [0, 1]. Figure 1 shows the distribution of the three groups of expansion terms. We can observe that the neutral terms are somehow isolated from the good and the bad terms to some extent (on the lower-right corner), but the good expansion terms are intertwined with the bad expansion terms.

This figure illustrates the difficulty to separate good and bad expansion terms according to term distributions solely. It is then desirable to use additional criteria to better select useful expansion terms.

4. Usefulness of Selecting Good Terms

Before proposing an approach to select good terms, let us first examine the possible impact with a good term selection process. Let us assume an oracle classifier that separate correctly good, bad and neutral expansion terms as determined in Section 3.

In this experiment, we will only keep the good expansion terms for each query. All the good terms are integrated into the new query model in the same way as either relevance model or mixture model. Table 2 shows the MAP (Mean Average Precision) for the top 1000 results with the original query model (**LM**), the expanded query models by the relevance model (**REL**) and by the mixture model (**MIX**), as well as by the oracle expansion terms (**REL+Oracle** and **MIX+Oracle**). The superscript, "*L*", "*R*" and "*M*" indicates that the improvement over LM, REL and MIX is statistically significant at p<0.05.

Models	**AP**	**WSJ**	**Disk4&5**
LM	0.2407	0.2644	0.1753
REL	0.2752^{L}	0.2843^{L}	0.1860^{L}
REL+Oracle	$\mathbf{0.3402^{R,L}}$	$\mathbf{0.3518^{R,L}}$	$\mathbf{0.2434^{R,L}}$
MIX	0.2846^{L}	0.2938^{L}	0.2005^{L}
MIX+Oracle	$\mathbf{0.3390^{M,L}}$	$\mathbf{0.3490^{M,L}}$	$\mathbf{0.2418^{M,L}}$

Table 2. The impact of oracle expansion classifier

We can see that the retrieval effectiveness can be much improved if term classification is done perfectly. The oracle expansion terms can generally improve the MAP of the relevance model and the mixture model by 18-30%. This shows the usefulness of correctly classifying the expansion terms and the high potential of improving the retrieval effectiveness by a good term classification. The MAP obtained with the oracle expansion terms represents the upper bound retrieval effectiveness we can expect to obtain using pseudo-relevance feedback. Our problem now is to develop an effective method to correctly classify the expansion terms.

5. Classification of Expansion Terms

5.1 Classifier

Any classifier can be used for term classification. Here, we use SVM. More specifically, we use the SVM variant *C-SVM* [2] because of its effectiveness and simplicity [20]. Several kernel functions can be used in SVM. We use the radial-based kernel function (RBF) because it has relatively fewer hyper parameters and has shown to be effective in previous studies [2],[5]. This function is defined as follows:

$$K(x_i, x_j) = exp\left[-||x_i - x_j||^2/2\sigma^2\right] \qquad (6)$$

where σ is a parameter controlling the shape of the RBF function. The function gets flatter when   is larger. Another parameter *C>0* in *C-SVM* should be set to control the trade-off between the slack variable penalty and the margin [2]. Both parameters are estimated with a 5-fold cross-validation to maximize the classification accuracy of the training data (see Table 7).

In our term classification, we are interested to know not only if a term is good, but also the extent to which it is good. This latter value is useful for us to measure the importance of an expansion term and to weight it in the new query. Therefore, once we obtain a classification score, we use the method described in [12] to transform it into a posterior probability: Suppose the classification score calculated with the SVM is *s*(*x*). Then the probability of *x* belonging to the class of good terms (denoted by **+1**) is defined by:

$$P(+1|x) = \frac{1}{exp[As(x)+B]} \qquad (7)$$

where *A* and *B* are the parameters, which are estimated by minimizing the cross-entropy of a portion of training data, namely the development data. This process has been automated in LIBSVM [5]. We will have *P*(**+1**|*x*)>0.5 if and only if the term *x* is classified as a good term. More details about this model can be found in [12]. Note that the above probabilistic SVM may have different classification results from the simple SVM, which classifies instances according to sign(*s*(*x*)). In our experiments, we have tested both probabilistic and simple SVMs, and found that the former performs better. We use the SVM implementation LIBSVM [5] in our experiments.

5.2 Features Used for Term Classification

Each expansion term is represented by a feature vector $F(e) = [f_1(e), f_2(e), \ldots, f_N(e)]^T \in \Re^N$, where T means a transpose of a vector. Useful features include those already used in traditional approaches such as term distribution in the feedback documents and term distribution in the whole collection. As we mentioned, these features are insufficient. Therefore, we consider the following additional features:

- co-occurrences of the expansion term with the original query terms;
- proximity of the expansion terms to the query terms.

We will explain several groups of features below. Our assumption is that the most useful feature for term selection is the one that makes the largest difference between the feedback documents and the whole collection (similar to the principle used in the mixture model). So, we will define two sets of features, one for the feedback documents and another for the whole collection. However, technically, both sets of features can be obtained in a similar way. Therefore, we will only describe the features for the feedback documents. The others can be defined similarly.

- **Term distributions**

The first features are the term distributions in the pseudo-relevant documents and in the collection. The feature for the feedback documents is defined as follows:

$$f_1(e) = log \frac{\sum_{D \in \mathcal{F}} tf(e,D)}{\sum_t \sum_{D \in \mathcal{F}} tf(t,D)}$$

where $\mathcal{F}$ is the set of feedback documents. $f_2(e)$ is defined similarly on the whole collection. These features are the traditional ones used in the relevance model and mixture model.

- **Co-occurrence with single query term**

Many studies have found that the terms that co-occur with the query terms frequently are often related to the query [1]. Therefore, we define the following feature to capture this fact:

$$f_3(e) = log \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{D \in \mathcal{F}} C(t_i, e|D)}{\sum_w \sum_{D \in \mathcal{F}} tf(w,D)}$$

where $C(t_i, e|D)$ is the frequency of co-occurrences of query term t_i and the expansion term *e* within text windows in document *D*. The window size is empirically set to be 12 words.

- **Co-occurrence with pairs query terms**

A stronger co-occurrence relation for an expansion term is with two query terms together. [1] has shown that this type of co-occurrence relation is much better than the previous one because it can take into account some query contexts. The text window size used here is 15 words. Given the set Ω of possible term pairs, we define the following feature, which is slightly extended from the previous one:

$$f_5(e) = log \frac{1}{|\Omega|} \sum_{(t_i,t_j) \in \Omega} \frac{\sum_{D \in \mathcal{F}} C(t_i, t_j, e|D)}{\sum_w \sum_{D \in \mathcal{F}} tf(w,D)}$$

- **Weighted term proximity**

The idea of using term proximity has been used in several studies [18]. Here we also assume that two terms that co-occur at a

Coll.	Percentage of good terms	SVM		
		Accuracy	Rec.	Prec.
AP	0.3356	0.6945	0.3245	0.6300
WSJ	0.3126	0.6964	0.3749	0.5700
Disk4&5	0.3270	0.6901	0.3035	0.5970

Table 3. Classification results of SVM

smaller distance is more closely related. There are several ways to define the distance between two terms in a set of documents [18]. Here, we define it as the minimum number of words between the two terms among all co-occurrences in the documents. Let us denote this distance between t_i and t_j among the set $\mathcal{B}$ of documents by $dist(t_i, t_j|\mathcal{B})$. For a query of multiple words, we have to aggregate the distances between the expansion term and all query terms. The simplest method is to consider the average distance, which is similar to the average distance defined in [18]. However, it does not produce good results in our experiments. Instead, the weighted average distance works better. In the latter, a distance is weighted by the frequency of their co-occurrences. We then have the following feature:

$$f_7(e) = log\frac{\sum_{i=1}^{n} C(t_i,e) dist\,(t_i,e|\mathcal{F})}{\sum_{i=1}^{n} C(t_i,e)}$$

where $C(t_i, e)$ is the frequency of co-occurrences of t_i and e within text windows in the collection: $C(t_i, e) = \sum_{D\in C} C(t_i, e|D)$. The window size is set to 12 words as before.

- **Document frequency for query terms and the expansion term together**

The features in this group model the count of documents in which the expansion term co-occurs with all query terms. We then have:

$$f_9 = \log\left[\sum_{D\in F} I\left(\left(\wedge_{t\in q}\, t \in D\right) \wedge e \in D\right) + 0.5\right]$$

where *I(x)* is the indicator function whose value is 1 when the Boolean expression *x* is true, and 0 otherwise. The constant 0.5 here acts as a smoothing factor to avoid zero value.

To avoid that a feature whose values varies in a larger numeric range dominates those varying in smaller numeric ranges, scaling on feature values is necessary [5]. The scaling is done in a query-by-query manner. Let $e \in GEN(q)$ be an expansion term of the query *q*, and $f_i(e)$ is one feature value of *e*. We scale $f_i(e)$ as follows:

$$f_i'(e) = \frac{f_i(e) - min_i}{max_i - min_i}, \text{ where } min_i = min_{e\in GEN(q)} f_i(e) \text{ and}$$

$$max_i = max_{e\in GEN(q)} f_i(e)$$

With this transformation, each feature becomes a real number in [0, 1].

In our experiments, only the above features are used. However, the general method is not limited to them. Other features can be added. The possibility to integrate arbitrary features for the selection of expansion terms indeed represents an advantage of our method.

5.3 Classification Experiments

Let us now examine the quality of our classification. We use three test collections (see Table 7 in Section 7.1), with 150 queries for each collection. We divide these queries into three groups of 50 queries. We then do leave-one-out cross validation to evaluate the classification accuracy. To generate training and test data, we use the method described in section 3 to label possible expansion terms of each query as good terms or non-good terms (including bad and neutral terms), and then represent each expansion with the features described in section 5.2. The candidate expansion terms are those that occur in the feedback documents (top 20 documents in the initial retrieval) no less than three times.

Table 3 shows the classification results. In this table, we show the percentage of good expansion terms for all the queries in each collection – around 1/3. Using the SVM classifier, we obtain a classification accuracy of about 69%. This number is not high. In fact, if we use a naïve classifier that always classifies instances into non good class, the accuracy (i.e. one minuses the percentage of good terms) is only slightly lower. However, such a classifier is useless for our purpose because no expansion term is classified as good term. Better indicators are recall, and more particularly precision. Although the classifier only identifies about 1/3 of the good terms (i.e. recall), around 60% of the identified ones are truly good terms (i.e. precision). Comparing to Table 1 for the expansion terms selected by the mixture model, we can see that the expansion terms select by the SVM classifier are of much higher quality. This shows that the additional features we considered in the classification are useful, although they could be further improved in the future.

In the next section, we will describe how the selected expansion terms are integrated into our retrieval model.

6. Re-weighting Expansion Terms with Term Classification

The classification process performs a further selection of expansion terms among those proposed by the relevance model and the mixture model respectively. The selected terms can be integrated in these models in two different ways: hard filtering, i.e. we only keep the expansion terms classified as good; or soft filtering, i.e. we use the classification score to enhance the weight of good terms in the final query model. Our experiments show that the second method performs better. We will make a comparison between these two methods in Section 7.4. In this section, we focus on the second method, which means a redefinition of the models $P(w|\theta_R)$ for the relevance model and $P(w|\theta_T)$ for the mixture model. These models are redefined as follows: For a term *e* such that $P(+1|e) > 0.5$,

$$P(e|\theta_R)^{new} = \frac{P(e|\theta_R)^{old}\left(1+\alpha P(+1|e)\right)}{Z}$$

$$P(e|\theta_T)^{new} = \frac{P(e|\theta_T)^{old}\left(1+\alpha P(+1|e)\right)}{Z} \qquad (8)$$

where Z is the normalization factor, and α is a coefficient, which is estimated with some development data in our experiments using line search [4]. Once the expansion terms are re-weighted, we will retain the top 80 terms with the highest probabilities for expansion. Their weights are normalized before being interpolated with the original query model. The number 80 is used in order to compare with the relevance model and the mixture model, which also use 80 expansion terms.

7. IR Experiments

7.1 Experimental Settings

We evaluate our method with three TREC collections, AP88-90, WSJ87-92 and all documents on TREC disks 4&5. Table 4 shows the statistics of the three collections. For each dataset, we split the available topics into three parts: the training data to train the SVM classifier, the development data to estimate the parameter α in equation 9, and the test data. We only use the title for each TREC topic as our query. Both documents and queries are stemmed with Porter stemmer and stop words are removed.

Name	Description	#Docs	Train Topics	Dev. Topics	Test topics
AP	Assoc. Press 88-90	24,918	101-150	151-200	51-100
WSJ	Wall St. Journal 87092	173,252	101-150	151-200	51-100
Disk4&5	TREC disk4&5	556,077	301-350	401-450	351-400

Table 4. Statistics of evaluation data sets

The main evaluation metric is Mean Average Precision (MAP) for top 1000 documents. Since some previous studies showed that PRF improves recall but may hurt precision, we also show the precision at top 30 and 100 documents, i.e., P@30 and P@100. We also show recall as a supplementary measure. We do a query-by-query analysis and conduct t-test to determine whether the improvement on MAP is statistically significant.

The Indri 2.6 search engine [17] is used as our basic retrieval system. We use the relevance model implemented in Indri, but implemented the mixture model following [22] since Indri does not implement this model.

7.2 Ad-hoc Retrieval Results

In the experiments, the following methods are compared:

LM: the KL-divergence retrieval model with the original queries;

REL: the relevance model;

REL+SVM: the relevance model with term classification;

MIX: the mixture model;

MIX+SVM: the mixture model with term classification.

These models require some parameters, such as the weight of original model when forming the final query representation, the Dirichlet prior for document model smoothing and so on. Since the purpose of this paper is not to optimize these parameters, we set all of them at the same values for all the models. Tables 5, 6 and 7 show the results obtained with different models on the three collections. In the tables, **imp** means the improvement rate over LM model, * indicates that the improvement is statistically significant at the level of $p<0.05$, and ** at $p<0.01$. The superscripts "R" and "M" indicate that the result is statistically better than the relevance model and mixture model respectively at $p<0.05$.

From the tables, we observe that both relevance model and mixture model, which exploit a form of PRF, can improve the retrieval effectiveness of LM significantly. This observation is consistent with previous studies. The MAP we obtained with these two models represent the state-of-the-art effectiveness on these test collections.

Comparing the relevance model and the mixture model, we see that the latter performs better. The reason may be the following: The mixture model relies more on the difference between the feedback documents and the whole collection to select the expansion terms, than the relevance model. By doing this, one can filter out more bad or neutral expansion terms.

On all the three collections, the model integrating term classification performs very well. When the classification model is used together with a PRF model, the effectiveness is always improved. On the AP and Disk4&5 collections, the improvements are more than 7.5% and are statistically significant. The improvements on the WSJ collection are smaller (about 3.5%) and are not statistically significant.

Model	P@30	P@100	MAP	Imp	Recall
LM	0.3967	0.3156	0.2407	-----	0.4389
REL	0.4380	0.3628	0.2752	14.33%**	0.4932
REL+SVM	*0.4513*	*0.3680*	*0.2959*^{R}	*22.93%***	*0.5042*
MIX	0.4493	0.3676	0.2846	18.24%**	0.5163
MIX+SVM	*0.4567*	*0.3784*	*0.3090*^{M,R}	*28.36%***	*0.5275*

Table 5. Ad-hoc retrieval results on AP data

Model	P@30	P@100	MAP	Imp	Recall
LM	0.3900	0.2936	0.2644	--------	0.6516
REL	0.4087	0.3078	0.2843	7.53%**	0.6797
REL+SVM	*0.4167*	*0.3120*	*0.2943*	*11.30%***	*0.6933*
MIX	0.4147	0.3144	0.2938	11.11%**	0.7052
MIX+SVM	*0.4200*	*0.3160*	*0.3036*^{R}	*14.82%***	*0.7110*

Table 6. Ad-hoc retrieval results on WSJ data

Model	P@30	P@100	MAP	Imp	Recall
LM	0.2900	0.1734	0.1753	-----------	0.4857
REL	0.2973	0.1844	0.1860	6.10%*	0.5158
REL+SVM	*0.2833*	*0.1990*	*0.2002*^{R}	*14.20%***	*0.5689*
MIX	0.3027	0.1998	0.2005	14.37%**	0.5526
MIX+SVM	*0.3053*	*0.2068*	*0.2208*^{M,R}	*25.96%***	*0.6025*

Table 7. Ad-hoc retrieval results on Disk4&5 data

“machine translation”			
Expansion terms	$P(t_i\|\theta_T)$	Expansion terms	$P(t_i\|\theta_T)$
compute	**0.0162**	year	0.0043
soviet	**0.0095**	work	0.0038
company	**0.0082**	make	0.0040
50	0.0074	***typewriter***	**0.0038**
english	**0.0072**	*busy*	0.0021
ibm	**0.0051**	increase	0.0021
people	0.0050	…..	….
“natural language processing”			
Expansion terms	$P(t_i\|\theta_T)$	Expansion terms	$P(t_i\|\theta_T)$
english	**0.0132**	publish	0.0041
word	**0.0092**	*nation*	0.0040
french	**0.0092**	**develop**	**0.0039**
food	0.0064	**russian**	**0.0038**
make	0.0050	***program***	**0.0037**
world	0.0047	***dictionary***	**0.0012**
gorilla	**0.0045**	……..	…..

Table 8. Expansion terms of two queries. The terms in *italic* are real good expansion terms, and those in **bold** are classified as good terms

About the impact on precision, we can also see that term classification can also improve the precision at top ranked documents, except in the case of Disk4&5 when SVM is added to REL. This shows that in most cases, adding the expansion terms does not hurt, but improves, precision.

Let us show the expansion terms for the queries "machine translation" and "natural language processing", in Table 8. The stemmed words have been restored in this table for better readability. All the terms contained in the table are those suggested by the mixture model. However, only part of them (in *italic*) is useful expansion terms. Many of them are general terms that are not useful, for example, "food", "make", "year", "50", and so on. The classification process can help identify well the useful expansion terms (in **bold**): although not all the useful expansion terms are identified, those identified (e.g. "program", "dictionary") are highly related and useful. As the weight of these terms is increased, the relative weight of the other terms is decreased, making their weights in the final query model smaller. These examples illustrate why the term classification process can improve the retrieval effectiveness.

7.3 Supervised vs. Unsupervised Learning

Compared to the relevance model and the mixture model, the approach with term classification made two changes: it uses supervised learning instead of unsupervised learning; it uses several additional features. It is then important to see which of these changes contributed the most to the increase in retrieval effectiveness.

In order to see this, we design a method using unsupervised learning, but with the same additional features. The unsupervised learning extends the mixture model in the following way:

Each feedback document is also considered to be generated from the topic model (to be extracted) and the collection model. We try to extract the topic model so as to maximize the likelihood of the feedback documents as in the mixture model. However, the difference is that, instead of defining the topic model $P(w|\theta_T)$ as a multinomial model, we define it as a log-linear model that combines all the features:

$$P(w|\theta_T) = \frac{1}{Z} exp[\lambda^T F(w)] \qquad (9)$$

where $F(w)$ is the feature vector defined in section 5.2, λ is the weight vector and Z is the normalization factor to make $P(w|\theta_T)$ a real probability. λ is estimated by maximizing the likelihood of the feedback documents. To avoid overfitting, we do regularization on λ by assuming that it has a zero-mean Gaussian prior distribution [2]. Then the objective function to be maximized becomes:

$$\mathcal{L}(\mathcal{F}) = \sum_{D\in\mathcal{F}} \sum_{w\in V} tf(w,D) \log((1-\alpha)P(w|\theta_C) + \alpha P(w|\theta T) - \sigma||\lambda||2 \qquad (10)$$

where σ is the regularization factor, which is set to be 0.01 in our experiments. α is the parameter representing how likely we use the topic model to generate the pseudo-relevant document. It is set at a fixed value as in [22] (0.5 in our case). Since $\mathcal{L}(\mathcal{F})$ is a concave function w.r.t. λ, it has a unique maximum. We solve this unconstrained optimization problem with L-BFGS algorithm [10].

Table 9 shows the results measured by MAP. Again, the superscript, "*M*" and "*L*" indicate the improvement over MIX and Log-linear model is statistically significant at p<0.05.

From this table, we can observe that the log-linear model outperforms the mixture model only slightly. This shows that an unsupervised learning method, even with additional features, cannot improve the retrieval effectiveness by a large margin. The possible reason is that the objective function, $\mathcal{L}(\mathcal{F})$, does not correlate very well with MAP. The parameters maximizing $\mathcal{L}(\mathcal{F})$ do not necessarily produce good MAP.

In comparison, the MIX+SVM model outperforms largely the log-linear model on all the three collections, and the improvements on AP and Disk4&5 are statistically significant. This result shows that a supervised learning method can more effectively capture the characteristics of the genuine good expansion terms than an unsupervised method.

Model	**AP**	**WSJ**	**Disk4&5**
MIX	0.2846	0.2938	0.2005
Log-linear	0.2878	0.2964	0.2020
MIX+SVM	0.3090^{M,L}	0.3036	0.2208^{M,L}

Table 9. Supervised Learning VS Unsupervised Learning

7.4 Soft Filtering vs. Hard Filtering

We mentioned two possible ways to use the classification results: hard filtering of expansion terms by retaining only the good terms, or soft filtering by increasing the weight of the good terms. In this section, we compare the two methods. Table 10 shows the results obtained with both methods. In the table, "M", "R", and "H" indicate the improvement over MIX, REL and HARD are statistically significant with p<0.05

From this table, we see that both hard and soft filtering improves the effectiveness. Although the improvements with hard filtering are smaller, they are steady on all the three collections. However, only the improvement over MIX model on the AP and Disk4&5 data is statistically significant.

In comparison, the soft filtering method performs much better. Our explanation is that, since the classification accuracy is far from perfect (actually, it is less than 70% as shown in Table 3), some top ranked good expansion terms, which could improve the performance significantly, can be removed by the hard filtering. On the other hand, in the soft filtering case, even if the top ranked good terms are misclassified, we will only reduce their relative weight in the final query model rather than removing them. Therefore, these expansion terms can still contribute to improving the performance. In other words, the soft filtering method is less affected by classification errors.

Model	**AP**	**WSJ**	**Disk4&5**
MIX	0.2846	0.2938	0.2005
MIX+HARD	0.2902^{M}	0.2989	0.2024^{M}
MIX+SOFT	*0.3090*^{M,H}	*0.3036*	*0.2208*^{M,H}
REL	0.2752	0.2843	0.1860
REL+HARD	0.2804	0.2864	0.1890
REL+SOFT	*0.2959*^{R,H}	*0.2943*	*0.2002*^{R}

Table 10. Soft Filtering VS Hard Filtering

7.5 Reducing Query Traffic

A critical aspect with query expansion is that, as more terms are added into the query, the query traffic, i.e. the time needed for its evaluation, becomes larger. In the previous sections, for the purpose of comparison with previous methods, we used 80 expansion terms. In practice, this number can be too large. In this section, we examine the possibility to further reduce the number of expansion terms.

In this experiment, after a re-weighting with soft filtering, instead of keeping 80 expansion terms, we only select the top 10 expansion terms. These terms are used to construct a small query topic model $P(w|\theta_T)$. This model is interpolated with the original query model as before. The following table describes the results using the mixture model.

Model	**AP**	**WSJ**	**Disk4&5**
MIX+SOFT-10	*0.2932*	*0.2915*	*0.2125*

Table 11. Soft filtering with 10 terms

As expected, the effectiveness with 10 expansion terms is lower than with 80 terms. However, we can still obtain much higher effectiveness than the traditional language model LM, and all the improvements are significantly significant.

The results with 10 expansion terms can also be advantageously compared to the mixture model with 80 expansion terms: for both AP and Disk4&5 collections, the effectiveness is higher than the mixture model. The effectiveness on WSJ is very close.

This experiment shows that we can reduce the number of expansion terms, and even with a reasonably small number, the retrieval effectiveness can be greatly increased. This observation allows us to control query traffic within an acceptable range, and make the method more feasible in the search engines.

8. Conclusion

Pseudo-relevance feedback, which adds additional terms extracted from the feedback documents, is an effective method to improve the query representation and the retrieval effectiveness. The basic assumption is that most strong terms in the feedback documents are useful for IR. In this paper, we re-examined this hypothesis on three test collections and showed that the expansion terms determined in traditional ways are not all useful. In reality, only a small proportion of the suggested expansion terms are useful, and many others are either harmful or useless. In addition, we also showed that the traditional criteria for the selection of expansion terms based on term distributions are insufficient: good and bad expansion terms are not distinguishable on these distributions.

Motivated by these observations, we proposed to further classify expansion terms using additional features. In addition, we aim to select the expansion terms directly according to their possible impact on the retrieval effectiveness. This method is different from the existing ones, which often rely on some other criteria that do not always correlate with the retrieval effectiveness.

Our experiments on three TREC collections showed that the expansion terms selected using our method are significantly better than the traditional expansion terms. In addition, we also showed that it is possible to limit the query traffic by controlling the number of expansion terms, and this still lead to quite large improvements in retrieval effectiveness.

This study shows the importance to examine the crucial problem of usefulness of expansion terms before the terms are used. The method we propose also provides a general framework to integrate multiple sources of evidence.

This study suggests several interesting research avenues for our future investigation: The results we obtained with term classification are much lower than with the oracle expansion terms. This means that there is still much room for improvement. In particular, improvement in classification quality could directly result in improvement in retrieval effectiveness. The improvement of classification quality could be obtained by integrating more useful features. In this paper, we have limited our investigation to only a few often used features. More discriminative features can be investigated in the future.

REFERENCES

[1] Bai, J. Nie, J., Bouchard, H. and Cao, G. Using query contexts in information retrieval. In the Proceedings of SIGIR'2007, Armsterdam, Netherlands, 2007.

[2] Bishop, C. Patten recognition and machine learning. Springer, 2006.

[3] Dempster, A. , Laird, N. and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B. 39(1):1-38, 1977

[4] Gao, J., Qi, H., Xia, X., and Nie, J. Linear discriminant model for information retrieval. In the Proceedings of SIGIR'2005, pp. 290-297, 2005.

[5] Hsu, C. Chang, C. and Lin, C. A practical guide to support vector classification. Technical Report, National Taiwan University.

[6] Joachims, T. Text categorization with support vector machines: learning with features. In ECML, pp.137-142, 1998.

[7] Kwok, K.L, Grunfeld, L., Chan, K., THREC-8 ad-hoc, query and filtering track experiments using PIRCS, In TREC10, 2000.

[8] Lavrenko, V. and Croft, B. Relevance-based language models. In the Proceedings of SIGIR'2001, pp.120-128, 2001.

[9] Metzler, D. and Croft, B. Latent Concept Expansion Using Markov Random Fields. In the Proceedings of SIGIR'2007, pp.311-318.

[10] Nocedal, J. and Wright, S. Numerical optimization. Springer, 2006.

[11] Peat, H.J. and Willett, P., The limitations of term co-occurrence data for query expansion in document retrieval systems. *JASIS*, 42(5): 378-383, 1991.

[12] Platt, J. Probabilities for SV Machines. Advances in large margin classifiers, pages 61-74, Cambridge, MA, 2000. MIT Press

[13] Robertson, S., and Sparck Jones, K. Relevance weighting of search terms. JASIST, 27:129-146, 1976

[14] Robertson, S.E., On term selection for query expansion, Journal of Documentation, 46(4): 359-364. 1990.

[15] Rocchio, J. Relevance feedback in information retrieval. In The SMART Retrieval System: Experiments in Automatic Document Processing, pages 313-323, 1971

[16] Smeaton, A. F. and Van Rijsbergen, C. J. The retrieval effects of query expansion on a feedback document retrieval system. *Computer Journal*, 26(3): 239-246. 1983.

[17] Strohman, T., Metzler, D. and Turtle, H., and Croft, B. (2004). Indri: A Language Model-based Search Engine for Complex Queries. In Proceedings of the International conference on Intelligence Analysis.

[18] Tao, T. and Zhai, C. An exploration of proximity measures information retrieval. In the Proceedings of SIGIR'2007, pp.295-302, 2007.

[19] Tao, T. and Zhai, C. Regularized estimation of mixture models for robust pseudo-relevance feedback. In the Proceedings of SIGIR'2006.

[20] Vapnik, V. Statistical Learning Theory. New York: Wiley, 1998

[21] Xu, J. and Croft, B. Query expansion using local and global document analysis. In the Proceedings of SIGIR'2006, pp.4-11, 1996.

[22] Zhai, C. and Lafferty, J. Model-based feedback in the KL-divergence retrieval model. In CIKM, pp.403-410, 2001a.

[23] Zhai, C. and Lafferty, J. A study of smoothing methods for language models applied to ad hoc information retrieval. In Proceedings of SIGIR'2001, pp.334-342, 2001b.

[24] Zhang, Y., Callan, J., The bais problem and language models in adaptive filtering. In the Proceedings of TREC11, pp.78-83, 2001