Informative term selection for automatic query expansion

Claudio Carpineto
Fondazione Ugo Bordoni, Rome
Italy
carpinet@fub.it

Renato De Mori
University of Avignon
France
renato.demori@
lia.univ-avignon.fr

Giovanni Romano
Fondazione Ugo Bordoni, Rome
Italy
romano@fub.it

Abstract Techniques for query expansion from top retrieved documents have been recently used by many groups at TREC, often on a purely empirical ground. In this paper we present a novel method for ranking and weighting expansion terms. The method is based on the concept of relative entropy, or Kullback-Lieber distance, developed in Information Theory, from which we derive a computationally simple and theoretically justified formula to assign scores to candidate expansion terms. This method has been incorporated into a comprehensive prototype ranking system, tested in the *ad hoc* track of TREC-7. The system's overall performance was comparable to median performance of TREC-7 participants, wich is quite good considering that we are new to TREC and that we used unsophisticated indexing and weighting techniques. More focused experiments showed that the use of an information-theoretic component for query expansion significantly improved mean retrieval effectiveness over unexpanded query, yielding performance gains as high as 14% (for non interpolated average precision), while a per-query analysis suggested that queries that are neither too difficult nor too easy can be more easily improved upon.

1. Introduction

Automatic query expansion from top retrieved documents is a well known retrieval strategy with clear potentials for addressing both thoretical limitations of information retrieval systems, such as the incapability of recovering from word mismatch between queries and documents, and practical deficiencies related to their usage in operational environments, such as the paucity of user-supplied query terms. While these potentials did not, historically, turn into actual better retrieval performance, due to losses in precision being higher than gains in recall, this is not the case in the TREC environment. The combination of better initial retrieval and the collection characteristics of TREC (longer and more numerous relevant documents than in the small test collections) makes this approach very successful (Buckley et al., 1995). According to Donna Harman (Harman, 1998), "by TREC-6 almost all groups were using variations on expanding queries using information from the top retrieved documents (pseudo-relevance feedback)".

The growing interest in this technique calls for a better understanding of its foundations and a more careful evaluation of its experimental design choices. The primary concern here is to develop well founded methodologies for ranking and weighing expansion terms, because most of the approaches that have been proposed leave something to be desired in terms of theoretical justification. Complementary, as virtually any proposed approach to query expansion relies on a number of parameters, it is important to study which factors are critical for good overall performance. Donna Harman (Harman 98), for instance, pointed out that, in the context of TREC, while there is general system convergence on some of the many parameters of query expansion needed for success, these still need to be tested by systems adopting these techniques.

Our participation in TREC-7, in the *Ad-hoc* track, was motivated by a desire to contribute to explore these issues. In particular, we were primarily interested in evaluating the retrieval effectiveness of a novel framework for query expansion based on ideas from Information Theory (Cover and Thomas, 1991). Additionally, we were concerned with evaluating the effectiveness of query expansion from top retrieved documents as the difficulty of the query varies.

2. Using information-theoretic "relative entropy" to select expansion terms

In order to discriminate between good expansion terms and poor expansion term it is convenient to assume that the differences between the distribution of terms in the overall document collection and the distribution of the same terms in a set of relevant documents are related to semantic factors. More precisely, we expect that good terms will occur with a higher frequency in relevant documents than in the whole collection, and poor terms will occur with the same frequency (randomly) in both.

To implement the view that the difference in the distribution of terms will reveal their likely relevance, we

need well founded and practical ways of comparing different distributions and assigning scores to terms based on such a comparison. Our approach is based on the relative entropy, or *Kullback-Leibler* distance (KD), between two distributions A and B. The relative entropy is customarily used to measure the extent of the error that we make by using A as a substitute for B. In our application we do not have a right distribution to approximate, thus it is more suitable to consider the sum of the difference between A and B and the difference between B and A. In the query expansion setting, the definition of this derived, symmetric distance becomes:

Let C be the set of all documents in the collection
Let V be the vocabulary of all the terms.
Let $t \in V$ be a word.
Let R be the set of top retrieved documents relative to a query.
Let v(R) be the vocabulary of all the terms in R.
Let $p_C(t)$ be the probability of $t \in V$ estimated using the whole collection. Let D_C be the corresponding distribution.
Let $p_R(t)$ be the probability of t estimated from the corpus R. Let D_R be the corresponding distribution. The *Kullback-Leibler* distance between the two distributions D_C and D_R is given by:

$$KD(D_C, D_R) = \sum_{1} \left\{ \left[p_R(t) - p_C(t) \right] \times \log \frac{p_R(t)}{p_C(t)} \right\} \quad (1)$$

The words to be considered for query refinement are those that mostly contribute to KD. In order to take into account the fact that it is possible that $v(R) \subset V$, a default probability is assumed for $p_R(t)$ when t does not appear in v(R), leading to the following definition:

$$p_R(t) = \begin{cases} \gamma \dfrac{f(t)}{NR} & \textit{if} \quad t \in v(R) \\ \delta\, p_C(t) & \textit{otherwise} \end{cases}$$

where f(t) is the frequency of t in R and *NR* is the number of terms in R. This scheme, in principle, better handles the sparse data problem when R is not sufficiently large. Since:

$$\sum_{t \in v(R)} f(t) = NR$$

the following relation must hold:

$$\gamma + \delta \sum_{t \notin v(R)} p_C(t) = \gamma + \delta A = 1$$

and KD can be rewritten as follows:

$$KD(D_C, D_R) = K_1 + K_2$$

$$K_1 = \sum_{t \in v(R)} \left\{ \left[\gamma \frac{f(t)}{NR} - p_C(t) \right] \log \frac{\gamma \dfrac{f(t)}{NR}}{p_C(t)} \right\}$$

$$K_2 = \sum_{t \notin v(R)} \left\{ \left[\delta\, p_C(t) - p_C(t) \right] \log \frac{\delta\, p_C(t)}{p_C(t)} \right\} =$$

$$= A(\delta - 1) \log \delta$$

As $\delta = \dfrac{1 - \gamma}{A}$

if: $m(\gamma) = \max_{t \notin v(R)} p_C(t) \dfrac{1 - \gamma - A}{A} \log \dfrac{1 - \gamma}{A}$

we may impose that selected terms for query refinement should respect the condition:

$$\left\{ \left[\gamma \frac{f(t)}{NR} - p_C(t) \right] \log \frac{\gamma \dfrac{f(t)}{NR}}{p_C(t)} \right\} > m(\gamma). \quad (2)$$

In other words, condition (2) states that the contribution of any selected term to KD should be greater than the contribution of every term not in v(R). As the left-hand side grows with γ while the right-hand side decreases with γ, it is always possible to find a value of $\gamma > 0$ such that the selected terms for query refinement do not contain any element not in v(S). This finding does not solve the parameter estimation problem but it supports using v(S) as an approximation of the set of candidate expansion terms. As γ does not influence the ranking of $t \in v(R)$, the following score can be used for ranking:

$$\sigma(t) = \left[\gamma \frac{f(t)}{NR} - p_C(t) \right] \log \frac{\gamma \dfrac{f(t)}{NR}}{p_C(t)} \quad (3)$$

with the first terms selected for query expansion. The same score can also be used for weighting the selected terms in the expanded query, in which case the result depends on the chosen value of γ.
For actual use in a retrieval system, a number of parameters must be chosen. This aspect is dealt with in the next section.

3. Description of our complete ranking methodology

1. *Text segmentation*. Our system first identified the

individual terms occurring in a text collection, ignoring punctuation and case.

2. *Word stemming*. To extract word-stem forms, we used a very large *trie*-structured morphological lexicon for English (Karp et al, 1992), that contains the standard inflections for nouns (singular, plural, singular genitive, plural genitive), verbs (infinitive, third person singular, past tense, past participle, progressive form), adjectives (base, comparative, superlative).

3. *Stop wording*. We used a stop list, contained in the CACM dataset, to delete from the texts common function (root) words. In addition, we removed the terms that appeared in more than 100,000 and less than 3 documents.

4. *Document weighting*. We assigned weights to the terms in each document by the classical *tf·idf* scheme.

5. *Weighting of unexpanded query:* To weigh terms in unexpanded query we used the function (log *tf*)·*idf*, where *tf* is the term frequency in the query and *idf* is the inverse document frequency.

6. *Document ranking with unexpanded query*: We computed an intermediate (or primary) document ranking by taking the inner product (with cosine normalization) between the document vectors and the unexpanded query vector.

7. *Expansion term ranking*: We used as set of candidate expansion terms the complete text of the first R retrieved documents. The candidates were ranked by using expression (3) with γ=1, which amounts to restricting the candidate set to the terms contained in R, and then the first E of them were chosen. To estimate $p_C(t)$, we used the ratio between the frequency of t in C and the number of terms in C, analogously to $p_R(t)$; in order to estimate $p_R(t)$ for the case when $t \in v(R)$, we used a more sophisticated function than f(t)/NR, taking also into account the likely degree of relevance of the documents retrieved in the initial run:

$$\frac{\sum_{d} f(t) \times score_d}{\sum_{t}\sum_{d} f(t) \times score_d} \qquad (4)$$

The argument made in Section 2 holds also for this new estimation function. It is also worth noting that the system selects only those terms with a higher estimated probability in the first retrieved documents than in the entire collection (i.e., such that the first factor in expression (3) is greater than zero); in fact, in our experiments the top terms always met this condition.

8. *Weighting of expanded query*. Expressed in vector space notation, $Q_{exp} = Q_{unexp} + \text{Smooth-Fn}\ (T_{exp})$, where T_{exp} contains the expansion terms weighted with their normalised σ-score, and Smooth-Fn is a smoothing function. The normalization was performed by dividing each score by the maximum score; the use of a smoothing function was due to the presence of a large fraction of suggested terms with very low scores. The unexpanded query was also normalized by the maximum possible weight.

9. *Document ranking with expanded query*: The final document ranking was computed by taking the inner product (with cosine normalization) between the document vectors and the expanded query vector.

The choice of the three parameters involved in our expansion method (number of pseudo relevant documents R, number of expansion terms E, and smoothing function Smooth-Fn) was based on earlier results obtained in past TREC conferences and on some preliminary experiments that we performed on the TREC-6 data. We selected two parameter combinations, (R=5, E=30, K=power 0.75) and (D=5, E=60, K=power 0.5), and computed the two corresponding document rankings (submitted as run "fub98a" and run "fub98b", respectively).

4. Computational efficiency

The whole system was implemented in Common Lisp and runs on a SUN-Ultra workstation. The time taken to index the whole collection (several hours) and to compute the primary ranking for each query (several seconds) was relatively large because I/O procedures were not optimized. Nonetheless, the time necessary to perform solely query expansion was negligible. As the collection frequencies were stored in the inverted file built from the set of documents, the computation of $p_C(t)$ was straightforward; to find $p_R(t)$, through expression (4), it was faster to perform one pass through the first retrieved documents. In fact, information-theoretic query expansion is practical even for interactive applications, provided that it is used in conjunction with an efficient ranking system.

5. Performance of expanded query versus unexpanded query

As our main goal was to evaluate the effectiveness of the information-theoretic expansion stage, we compared the performance of document ranking with unexpanded query

with that of the two document rankings with expanded query. The results are shown in Figure 1 and Table 1, using the standard TREC performance evaluation measures.

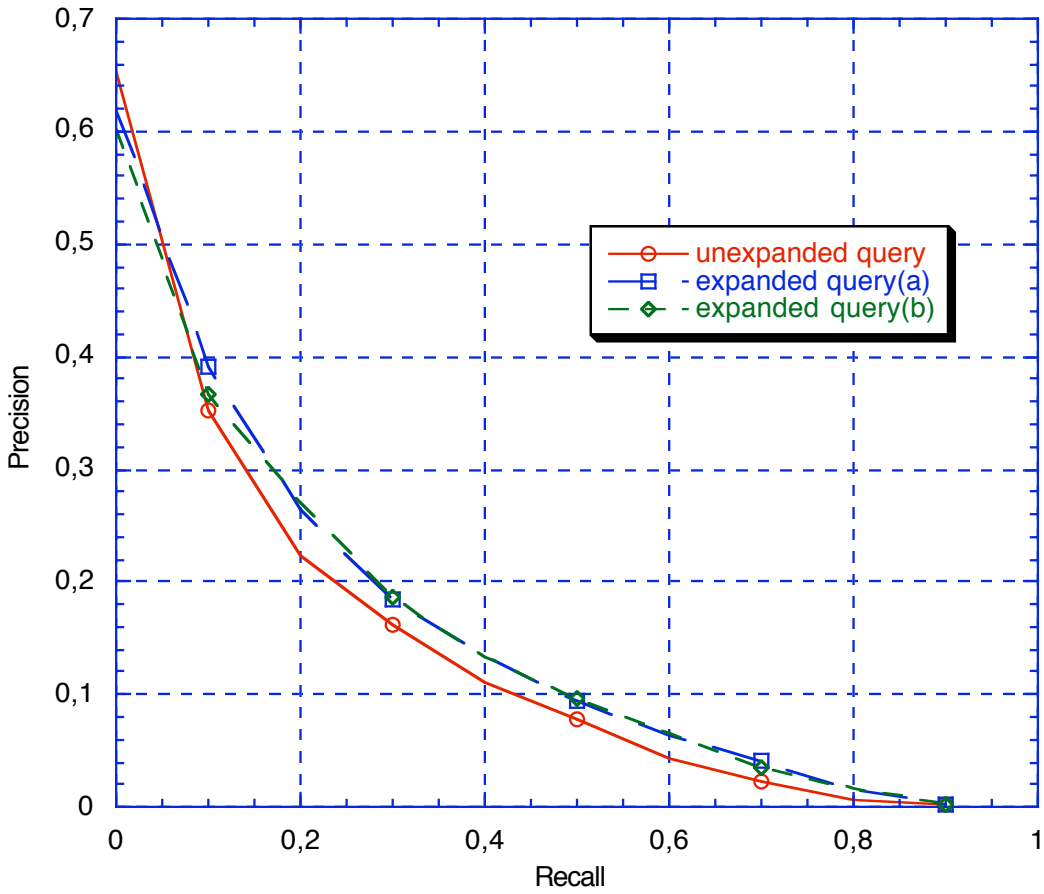

Figure 1. Comparative performance of ranking with and without query expansion (interpolated recall-precision curve).

It turned out that the two rankings with expanded query achieved very similar retrieval effectiveness, and that both of them had better performance than ranking without expansion at almost all evaluation points, with one main exception at recall=0 in the interpolated recall-precision curve. In particular, when passing from unexpanded query to expanded-query-(a), the overall number of relevant retrieved documents increased by 11.92%, average precision by 14.46%, and R-precision by 8.62%. The performance improvement was therefore apparently consistent in all of these tasks; in fact, the benefit of our expansion scheme was statistically significant in all of these measures.

It is also useful to compare the overall performance of our system with that of the other official runs in the Ad-hoc category. For instance, with respect to the the total number of retrieved relevant documents, our run (fub98a) achieved better than median performance for 17 topics, median performance for 4 topics, and worse than median performance for 29 topics. For the other evaluation measures, the behavior was similar. The overall performance of our system was therefore relatively good on average, especially considering that we are new to TREC, but it was definetely inferior to that of the best TREC systems. This implies that the utilization of the query expansion mechanism, while resulting in a marked improvement over ranking with unexpanded query, was not sufficient alone to compensate for the limited effectiveness of the primary ranking scheme.

In fact, it is the combination of several ingredients that makes systems successful at TREC. The current version of our system performs very conservative stemming,