Integrating Logical Operators in Query Expansion in Vector Space Model

Jian-Yun Nie, Fuman Jin
DIRO, Université de Montréal
C.P. 6128, succursale Centre-ville, Montreal
Quebec, H3C 3J7 Canada
{nie, jinf}@iro.umontreal.ca

Abstract

Query expansion is an effective way to extend the coverage of retrieval to the related documents. Various approaches have been proposed and many of them are based o vector space model. The expansion process consists of simply adding expansion terms into the original vector. In this paper we argue that this simplistic expansion method can bias the focus of the original query, because the expanded terms add additional emphasis to the original term. Instead of adding expansion terms into the vector, we propose to combine them with the original terms by means of a logical OR operator. In this way, the expansion terms are considered as alternatives to the original terms, and the focus of the whole query remain the unchanged. Our experiments on a REC collection show that this expansion approach is more effective than simple addition approach.

1. Introduction

One of the problems in IR is that queries are often a very partial specification of the information need. This is especially the case in the Web environment. Query expansion has often been suggested as a solution to this problem. Query expansion tries to extend the coverage of the retrieval to the related documents that do not necessarily contain the same words as those of the query. In the previous studies on query expansion, much emphasis has been put on the selection of expansion terms, either through a manually constructed thesaurus [Voorhees 93], a statistical measure based on co-occurrences or a combination of them [Mandala 99].

Te previous studies have shown various results with query expansion. Voorhees showed that using Wordnet [Miller 90] for query expansion in this way, the IR effectiveness is rather decreased. In the work of Mandala, the relations stored in Wordnet are combined with Mutual Information (MI) and another similarity measure based on syntactic dependency. He shows that such a combination can greatly improve IR effectiveness.

However, in all these studies, it has been taken as granted that the expanded terms should be added into the original vector directly. For example, if the term A is a related term to B, than a query containing B can be expanded by adding A into the query vector. No question has been raised concerning the appropriate way to integrate expansion terms into the query vector.

In this study, we propose another way to integrate expansion terms into query vector. Our basis idea is that an expansion term represents an alternative expression of the original term. Therefore, it should be combined with the original term by the logical relation OR.

In the following section, we will further motivate our approach. In section 3, this approach is compared with the direct addition approach on a TREC test collection. Our experiments will show that our expansion approach is more appropriate.

2. Expansion terms as alternatives

2.1. Principle of query expansion

Let us firs give some more example of the direct addition approach to query expansion. Suppose that an original query contains two terms A0 and B0. Through a resource (either a manual thesaurus or a statistical calculation), we know that A0 is strongly related to A1, A2, A3 and A4, and no related term is known for B0. If all the related terms are added into the query vector through the query expansion process, then the new vector will contain 6 terms or 6 non-empty dimensions. 5 of them are related to the aspect A and one to the aspect B. We have to notice that all the dimensions in this vector are treated in the same way. For example, the inner product similarity will simply sum up the similarities of a document to every dimension. We cab easily see that a document will have 5 more chances to satisfy the aspect A than the aspect B.

Now let us consider the following two cases.

1. The expansion terms are true synonyms that have little chance to co-occur with the original query term and with ach other.

   In this case, one may consider that no document will simultaneously contain more than one of the terms A0, A1, …, A4. Then in practice, the expanded query vector is equivalent to a set of alternative vectors: (A0, B0), (A1, B0), …, (A4, B0). In this particular case, no particular problem arises from the direct addition approach.

2. The expansion terms are related terms that may co-occur with the original term and with each other.

   This case occurs much more often in the previous experiments in query expansion. In fact, few studies have limited query expansion to the true synonymy relation. This is due to the fact that the true synonymy relation can only expand a query with a very limited number of additional terms, and it is difficult to guarantee that all the expansion terms are true synonyms (unless they have been selected rigorously). Even if we use resources such as Wordnet, we cannot guarantee that no synonym will co-occur with the original term. More often, query expansion exploits statistical relationships based on word co-occurrences. The expansion terms are more related ones than true synonyms. They are highly probable to co-occur in the same document as the original query term. If we return to our example, this means that a document will likely contain A0, A1, … and A4. If all the occurrences of these terms are considered and their similarities are summed up as the correspondence of the document to the query with respect to the A aspect, then the A aspect will take much higher importance than the aspect B that has not been expanded. In other words, in the expanded query vector, the aspect that has been more expanded will be implicitly attributed higher importance. This potentially will create a bias to the original information need. For example, suppose that the original query contains two terms with equal importance: A0 = “computer” and B0 = “graphics”; and suppose the expansion terms are A1 = “calculator”, A2 = “data processor”, A3 = “electronic computer” and A4 = “information”, the expanded query will contain the following terms:

```
(computer, graphics, calculator, data_processor,
 electronic_computer,  information_processing).
```

One can clearly see that much higher emphasis has been put on "computer" than on "graphics".

The above example shows the possible bias that query expansion may create. The bias is due to the unbalanced numbers of expansion terms for each query term. Even if one tries to take the number of expansion terms into account (e.g. by dividing the weight of each expansion term by the number of the expansion terms), this problem cannot be solved appropriately. In fact, such a division can solve the unbalance problem in the second case shown above, but will create additional problem in the first case: if a document only contains one of the many true synonyms, its similarity to this aspect is not fully considered.

The fundamental problem in the above approach is the following inconsistence: On one hand, when query is expanded, all he expansion terms are considered to be dependent to the original query term. On the other hand, when the expanded query is evaluated, all the dimensions are considered independently. This contradiction is the basic problem of the simple addition approach.

A radical solution lies in the creation of a more appropriate relationship between the original query term and the expanded terms in the expanded vector. That is, when the expansion terms are added because they are related to an original term, this relation should also be taken into account during the evaluation process.

Intuitively, when a query term is expanded with another term, the later should be considered as an alternative expression of the former. Therefore, the most natural logical operator to connect them is OR. We therefore suggest a vector representation that integrate the logical OR relation.

For our earlier example, the expanded query would rather be:

```
(computer OR calculator OR data_processor OR
   electronic_computer OR  information_processing,
 graphics).
```

The first composed element represent one single dimension in this vector. This dimension may be expressed in 5 alternative ways. The whole vector contains two dimensions, as in the original query. The relative importance between them is unchanged. The only change we bring into the new vector is the way to evaluate the first dimension.

A more general form of expanded query would contain a weight associated to each term, such as follows:

(computer^{w1} OR calculator^{w11} OR data_processor^{w12} OR
electronic_computer^{w13} OR information_processng^{w14},
graphics^{w2}).

The weights w1 and w2 are those in the original query that can be determined either by the user or with a tf*idf weighting schema. We will come back later on the problem of determining the weights w11, …, w14 for the expansion terms. For the moment, let us assume that such an expanded query has been created, and examine the problem of its evaluation.

2.2. Evaluation of an expanded query

Two problems have to be dealt with for the evaluation of an expanded query:

1. How to evaluate the OR relation?
2. How to integrate this logical operator in a vector space model?

Because of the uncertainty nature of query evaluation, the classical evaluation of OR cannot be used here. For the evaluation of OR in our case, we can use the evaluations proposed in the fuzzy set theory. The two following forms of evaluation of OR are among the most often used (where W is a fuzzy function which corresponds in our case to the similarity of a document to a query A OR B or part of it):

$$W(A \text{ OR } B) = \max(W(A), W(B)) \quad (1)$$

$$W(A \text{ OR } B) = W(A) + W(B) - W(A)*W(B) \quad (2)$$

The choice of an appropriate formula may be done though experiments. In our preliminary tests, it turns out that the second formula performs better than the first one.

The next question is how to integrate these evaluations into the calculation of similarity in vector space model. There are to solutions:

- We can transform the expanded query into a logical combination (with OR) or vectors. For example, the expanded query (A OR B, C) is equivalent to (A, C) OR (B, C).
- We can also directly evaluate each dimension of the expanded query. For a dimension corresponding to a logical expression, a singe similarity value will be calculated. The calculation of this value will take into account the correspondence of all the alternative terms with the document and the logical relation between them.

The first implementation is inefficient. In fact, as the term C may repeat in several elements in the disjunction, it has to be evaluated several times. Therefore, we implement the evaluation in the second way.

Our implementation also makes use of inverted file. We choose to use inner product as the formula of similarity. The following algorithm is used to obtain the resulting set of weighted documents for an Expanded query Exp_query:

```
eval(Exp_query)
    list = ∅;
    For each dimension_i in Exp_query:
          doc_list_i = eval_d(dimension_i);
          list = list ⊕ doc_list_i;
    return list;

eval_d(dimension)
    list = ∅;
    for each disjuncted term t_j^wj in dimension:
          list_j = list of documents containing t_j, the weight of
                   each document is multiplied by w_j;
          list = list ∇ list_j;
    return list;
```

In this algorithm the operator $\oplus$ denotes the union of two lists of weighted documents by summing up the weights of the common elements in the two lists. This is the operator corresponding to the classical evaluation in vector space model. Te operator ∇ is a combination of two lists of weighted documents by combining the weights of common elements according to one of the evaluation formulas for OR.

It is interesting to notice that the above algorithm is a generalization of the classical inner product evaluation of vector space model. To reproduce the classical evaluation, it is sufficient to use the following evaluation for OR:

$$W(A \text{ OR } B) = W(A) + W(B) \quad (3)$$

2.3. Determining related terms and their weights

Another important question in query expansion is how to determine the related terms to be used and their weights. In several studies (e.g. [Voorhees 93]) Wordnet is used to provide related terms. It turns out that using these terms to expand queries do not help in IR evaluation. Our preliminary tests also confirmed this fact. In [Mandala 99], Wordnet I combined with statistical measures between words based on their co-occurrences. It is shown that this combination greatly improves IR effectiveness. However, as all the measures have been tested in combination, it is not clear how each of the factors between Wordnet and statistical measures contributed in the evaluations.

In our experiments, we compared the use of Wordnet with that of statistical relations based on Mutual Information (MI). The latter performed much better. Therefore, in this paper, we only report the use of MI. Mutual information is defined as follows:

$$MI(x, y) = P(x, y) \times \log \frac{P(x, y)}{P(X) \times P(y)}$$

where x and y are terms, $P(x)$ is the probability of x determined by

$P(x)$ = #doc. containing x / #doc. of the collection;

$P(x,y)$ is the probability that x and y co-occur in the document:

$P(x,y)$ = #doc. containing x and y / #doc. of the collection.

The values of MI vary greatly from term to term. In order to make MI more comparable, we use the following normalized MI:

$$NMI(x, y) = MI(x, y) / \max(MI(x, y_i))$$

where x is an original query term, and y_i any possible expansion term.

As the expansion terms determined by MI are usually only related to some degree to the original information need, it is reasonable to assign a lower weight to them than to the original query terms. Therefore, an additional coefficient C ($0 \le C \le 1$) is added to decrease the importance of expansion terms. The weight of an expansion term y for the original query term x is then determined as follows:

$$W_x(y) = C * NMI(x, y)$$

Several values of C will be tested in our experiments described in the next section.

4. Experiments

We tested the methods on the AP collection used in TREC. This collection contains ?? documents in English. ?? queries are provided with standard answers. We use a modified version of Smart [Buckley 85] for our experiments.

We notice that terms that have a marginal MI with original query terms are usually noise that are not related to the query. They happen to co-occur with query terms in some documents. It is better to eliminate them in query expansion. Therefore, we also impose a fixed number *N* of expansion terms for each query term.

The following table shows the results we have obtained so far. The numbers are percentages of improvement with query expansion over the standard vector evaluation without query expansion.

	Number of expansion terms *N*									
Exp. method	2	5	10	15	20	25	30	50	60	90
Formula (3) *C*=0.25	1.4	7.2	9.2	10.1	2.8	3.3	3.2	3.9	2.4	2.6
Formula (2) *C*=0.25	1.5	7.8	10.7	11.3	11.4	10.6	9.3	2.3	1.1	-3.7
Formula (2) *C*=0.20	1.6	5.8	8.9	10.0	9.0	(?)	9.1	4.9	5.0	5.3
Formula (2) *C*=0.30	1.1	7.1	8.9	3.7	2.3	(?)	-0.9	-0.1	0.1	-1.3

Table 1. Experimental results with query expansion

As we can see, when a reasonable number of query expansion terms are used, query expansion usually increases IR effectiveness. The best number of expansion terms seems to be between 10-20.

If we compare the proposed expansion method that integrates logical OR (Formula (2), C=0.25), with simple addition approach (Formula (3), C=0.25), we can see that the former performs slightly better. In particular, when the number of expansion terms increases to more than 20, there are larger differences between them. This result shows that Formula (2) may be more appropriate to query expansion than Formula (3), which has been used in most other studies. This is the main conclusion of this study. However, this conclusion is preliminary based on the above partial results. We expect to obtain stronger conclusions once all the test results are obtained.

We also tested a few variations of the coefficient *C*. It turns out that this coefficient has a great impact on IR effectiveness. Among the three values tested so far, 0.25 seems to produce the best result.

We mentioned that Formula (2) produces better results than Formula (1) in our preliminary tests. However, the test condition is slightly different from that of the above table. So we do not show the comparison now. Our tests are still ongoing, and we expect to obtain additional results in the coming days, and report them in the final version of this paper.

5. Related work

Our approach to query expansion is different from most previous studies. In this study, we argue that an appropriate combination of the expansion terms with the original terms is an important problem to deal with in query expansion. I the previous studies, it has been taken as granted that expansion terms should be added as additional dimensions in the resulting vector (e.g. in [Voorhees 93] and [Mandala 99]). Our preliminary results seem to support this claim.

The work is also comparable to the tentative of [Wong 85] that tries to create more complex relationships within vectors. Wong et al. observed that the underlying independence assumption in vector representation is not reasonable. They suggest to consider dependencies between dimensions in a Generalized Vector Space Model proposed. However, the method of Wong et al.

suffers from the complexity problem. In practice, it is difficult to fully implement it. Out method does not suffer from this problem. It is a method that can be efficiently implemented in practice.

In fact, in our approach, we have implicitly assumed independence among original query terms. This means that each word in a query represents a different aspect of the information need. This assumption is reasonable, in particular, when queries are short. In short queries, the same aspect usually does not repeat. However, when queries are expanded, we can no longer assume that the expansion terms are also independent from the original terms and from each other among them. Our present study proposes to expand each dimension independently, and an expanded dimension is represented as a logical disjunction.

This work is also related to the Extended Boolean Model proposed by Salton et al. [Salton 83]. In this study on p_norm model, they show that the relationship between different dimensions in a vector is indeed a neutralized logical relation (i.e. no distinction between AND and OR). In our study, we propose a stronger OR relation to replace the neutralized logical relation to connect expansion terms to the original query terms.

In this study, each dimension is expanded independently. A more reasonable approach would be considering the expansion in dependence with the whole query. That is to consider other query terms as a context to help determine the most appropriate expansion terms. Such an expansion process is in the same line as the work by Qiu and Frei [Qiu 93]. We are currently implementing such an expansion process.

6. Conclusions

Query expansion is considered as an effective way to extend the coverage of a query in order to find strongly related documents. In the previous studies using vector space model, expansion terms have been usually directly added into the vector as additional dimensions. As dimensions are assumed to be independent, this simple addition approach can greatly alter the emphasis of the original query.

In this study, we proposed to use logical OR to connect expansion terms to the original query terms. The idea is to consider expansion terms as alternative expressions of the original terms.

Our experimental tests suggest that this new expansion method is more appropriate than the simple addition approach.

We are currently undertaking additional tests. More test results will be reported in a later version of this paper.

References

Buckley, C. *Implementation of the SMART information retrieval system*, Technical report, #85-686, Cornell University, 1985.

Mandala, R., Tokunaga, T. Combining multiple evidence from different types of thesaurus for query expansion, *Development on Information Retrieval - ACM-SIGIR*, 1999, pp. 191-197.

Miller, G., Wordnet: an on-line lexical database, in *International Journal of Lexicography*, vol. 3, 1990.

Qiu, Y., Frei, H.-P. Concept based query expansion, *Research and Development on Information Retrieval - ACM-SIGIR*, 1993, pp. 160-169.

Salton, G., Fox, E. A., and Wu, H. Extended Boolean information retrieval. *Commun. ACM*, 26(11):1022--1036, Nov. 1983.

Voorhees, E. M. Using Wordnet to disambiguate word senses for text retrieval. *Research and Development on Information Retrieval - ACM-SIGIR*, Pittsburgh, 1993, pp. 171-180.

Wong, S.K.M., Ziarko, W., Wong P.C.N., Generalized vector space model in information retrieval, *Research and Development on Information Retrieval - ACM-SIGIR*, 1985, pp. 18-25.