UCM-Y!R at CLEF 2008 Robust and WSD tasks

José R. Pérez-Agüera and Hugo Zaragoza
University Complutense of Madrid and Yahoo! Research
jose.aguera@fdi.ucm.es, hugoz@yahoo-inc.com

Abstract

We explore the use of state of the art query expansion techniques combined with a new family of ranking functions which can take into account some semantic structure in the query. This structure is extracted from WordNet similarity measures. Our approach produces improvements over the baseline and over query expansion methods for a number of performance measures including GMAP.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval

General Terms

Natural Language Processing, Information Retrieval, Robust Retrieval

Keywords

Query expansion, semantic expansion

1 Introduction

Exploiting semantic information for information retrieval is known to be very hard. One of the problems, in our opinion, is the term independence hypothesis. A second problem is that of query-dependant semantics: two terms semantically related in a query may not be so in the next. We try to address these two problems. We propose to make explicit some of the term dependence information using a form of structured query (which we call query clauses), and to use a ranking function capable of taking the structure information into account. We combine the use of query expansion techniques and semantic disambiguation to construct the structured queries that are both semantically rich and focused on the query.

1.1 Ranking Function

Our baseline will be the BM25 ranking function[10]:

$$R(q,d) = \sum_{t\ en\ q} \frac{tf_t^d}{k_1((1-b)+b*\frac{l_d}{avl_d})+tf_t^d} * idf(t) \quad (1)$$

$$idf(t) = \frac{N - df(t) + 0.5}{df(t) + 0.5} \quad (2)$$

For all our experiments we used Lucene[1], modifying the raking functions as needed.

[1] http://lucene.apache.org

1.2 Query Expansion Algorithms

Our first approach is to apply state of the art query expansion methods. We selected two methods which we find very effective.

1.3 Information-theoretic approach

One of the most interesting approaches based on term distribution analysis has been proposed by C. Carpineto et. al. [2], and uses the concept the Kullback-Liebler Divergence [3] to compute the divergence between the probability distributions of terms in the whole collection and in the top ranked documents obtained for a first pass retrieval using the original user query. The most likely terms to expand the query are those with a high probability in the top ranked set and low probability in the whole collection. For the term t this divergence is:

$$w(t) = P_R(t)\log\frac{P_R(t)}{P_C(t)} \tag{3}$$

where $P_R(t)$ is the probability of the term t in the top ranked documents, and $P_C(t)$ is the probability of the term t in the whole collection.

1.4 Divergence From Randomness for Query Expansion

The Divergence From Randomness (DFR) [1] term weighting model infers the informativeness of a term by the divergence between its distribution in the top-ranked documents and a random distribution. The most effective DFR term weighting model is the *Bo1 model* that uses the Bose-Einstein statistics [8, 6]:

$$w(t) = \mathrm{tf}_x \log_2(\frac{1+P_n}{P_n}) + \log(1+P_n) \tag{4}$$

where tf_x is the frequency of the query term in the x top-ranked documents and P_n is given by $\frac{F}{N}$, where F is the frequency of the query term in the collection and N is the number of documents in the collection.

We have used the first document retrieved for term extraction. The number of terms used to expand the original query has been 40.

1.5 Methods for Reweighing the Expanded Query Terms

After the list of candidate terms has been generated by one of the methods described above, the selected terms which will be added to the query must be re-weighted. Different schemas have been proposed for this task. We have compared these schemas and tested which is the most appropriate for each expansion method and for our combined query expansion method.

The classical approach to term re-weighting is the Rocchio algorithm [11]. In this work we have used Rocchio's beta formula, which requires only the β parameter, and computes the new weight qtw of the term in the query as:

$$qt_w = \frac{qtf}{qtf_{max}} + \beta\frac{w(t)}{w_{max}(t)} \tag{5}$$

where $w(t)$ is the original expansion weight of term t, $w_{max}(t)$ is the maximum $w(t)$ of the expanded query terms, β is a parameter, qtf is the frequency of the term t in the query and qtf_{max} is the maximum term frequency in the query q. In all our experiments, β is set to 0.3.

1.6 Query Performance Prediction

Query expansion is known to degrade the performance of some queries. In order to alleviate this problem we experimented with query quality predictors [4]. For efficiency reasons we only consider pre-retrieval methods. In particular, we used the AvICTF predictor proposed by [5]:

$$AvICTF = \frac{\log_2 \prod_{t \in Q} ICTF}{ql} = \frac{\log_2 \prod_{t \in Q} \frac{T}{F}}{ql} \tag{6}$$

where T is the number of tokens in the collection. The predictor is used as follows. We compute the AvICTF value of the expanded query. If this value is above a certain threshold (9.0) we will use the expanded query. Otherwise we use the original query. The threshold was found empirically on the training corpus.

2 Standard Query Expansion Results

In table 1 we report results on the baseline and query expansion for several evaluation measures, including GMAP which is perhaps the most interesting in the context of robust retrieval.

	MAP	GMAP	R-PREC	P@5	P@10
BM25 (baseline)	.3614	.1553	.3524	.4325	.3663
BM25 + KLD	.3833	.1527	.3647	.4575	.3869
BM25 + Bo1	.3835	.1528	.3615	.4613	.3844
BM25 + Bo1 + AvICTF	.3811	.1518	.3587	.4550	.3831

Table 1: Evaluation for different expansion methods.

As we can see, the query expansion methods obtain some improvement over the baseline, for all linear average measures, but not for GMAP. As it is usually the case, the query expansion methods are hurting the performance of the hardest queries. AvICTF it is helping somewhat but not enough to improve over the baseline.

3 Structured Query Expansion

Simply adding terms to a query may not be the best way to enrich them. We believe that adding related terms worsens the term independence hypothesis. In this section we explore an alternative family of ranking functions that addresses this issue. This scoring method is inspired in the fielded version of BM25 proposed in [9], and was proposed in [7], where it is described in more detail. Here we will give only a brief description.

Related terms are grouped in sets called clauses, and queries are defined as sets of clauses. Terms within the clauses and clauses themselves may be weighted. Each clause is considered as a pseudo term with each own tf and idf:

$$score(d, qc) = \sum_{i=1}^{n} \frac{tf(c_i, d)}{k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl}) + tf(c_i, d)} \cdot Eidf(c_i, d) \tag{7}$$

where qc is the expanded query with clauses, c_i is the ith clause, and $tf(c_i, d)$ is the sum of term frequencies in the clause[2]:

$$tf(c_i, d) = \sum_{t \in c_i} tf(t, d)$$

[2]In the original formulation [7] $tf(c_i, d)$ was defined as a weighted sum, taking w_t into account. This is more general and makes more sense from a theoretical point of view. However, for this task we observed better results if we dropped the weights. They are only used in the computation of $Eidf$. This needs further investigation.

We replace the *idf* term by a *clause idf*, defined as the expected *idf* of a term in the clause:

$$Eidf(c,d) = \frac{1}{\sum_{t' \in c} w_{t'} \cdot tf(d,t')} \sum_{(t,w_t) \in c} w_t \cdot tf(d,t) \cdot idf(t) \quad (8)$$

This has several nice properties, for example terms added with very small weights have very small effect on the clause *idf*, and terms not occurring in the document have no effect at all. See [7] for more details and a comparaison to other solutions. However, it has the dissadvantage that it needs to be computed for every clause in every document, at query time, unlike *idf* which can be pre-computed for each term.

This scoring method provides a method to introduce into the ranking function the expanded terms without the need of Rocchio. The question remains how to construct the clauses. Our hypothesis is that semantically related terms should be grouped in clauses. The CLEF corpus is ideal to test this hypothesis since all the terms in it have be annotated with their corresponding synset in WordNet. Our approach is described in the next section.

3.1 Exploiting WSD and WordNet for the construction of clauses

We need to construct a structured query from the initial user query. We have explored many possibilities, most of which lead to bad results (worse than simple expansion). We describe here the method used for our CLEF submission, which was quite successful.

In our experience state of the art query expansion methods are superior than *semantic expansion* methods based on WordNet or corpus statistics; their main advantage is that they lead to expansions that are truly query dependant; semantic information tends to be too vague and it is hard to use without knowing the context in which a word is used. However, the idea behind our method is that query expansion and semantic information may be used complementary. In particular, semantic information may be useful to decide the semantic *query structure* (query clauses in our case).

We proceed as follows. First we assign to each original query term a different query clause (we assume query terms to be independent in the traditional way). We assign the weight of 1 to these terms. Then we do standard query expansion and select the usual number of expansion terms (40 in our case) for the query. We use the DFR Bo1 method for this, although similar results can be obtained with the other methods. We then compute a semantic similarity between each original query word and each expansion word (discussed below). If this similarity is above a threshold α, we include the expanded term in the clause of the original term; we assign to this term a weight equal its expansion weight. All the expanded terms remaining are grouped into an extra query clause. This way, the number of clauses is always equal to $|q| + 1$.

As an example, let's say that the original query was a, b and the terms c, d, e were found to be good expansion terms, with weights w_c, w_d, w_e respectively. After computing the 6 semantic distances between original and expanded terms, we would check which were above a threshold α. Say for example that only d was sufficiently similar to b, all other similarities being bellow α. Then we would end up with the query:

$$\{\ \{(a,1)\},\ \{(b,1),(d,w_d)\},\ \{(c,w_c),(e,w_d)\}\}$$

Semantic similarities are computed based on WordNet. There exists an extensive literature on measures of semantic similarity. We have used the WordNet Similarity[3] package, which contains many semantic measures. In particular we used (after some experimentation) the *wup* measure [12] which is based on the LCS (Lexical Conceptual Structure) depth of the term pair in WordNet. The threshold α is a free parameter and we submitted results with different thresholds. In order to map the terms to WordNet, we used the WSD information in the corpus.

3.2 Results

We report here results on the semantic clause method described above.

[3] http://wn-similarity.sourceforge.net/

Table 2: Results for clause queries using different similarity thresholds in WordNet. α is the similarity threshold.

	MAP	GMAP	R-PREC	P@5	P@10
BM25 (baseline)	.3614	.1553	.3524	.4325	.3663
BM25 + Bo1	.3835	.1528	.3615	.4613	.3844
BM25 + Bo1 + Clauses ($\alpha = 0.0$)	.3937	**.1620**	.3735	.4600	.3869
BM25 + Bo1 + Clauses ($\alpha = 0.3$)	.3935	.1613	.3726	.4563	.3869
BM25 + Bo1 + Clauses ($\alpha = 0.6$)	.3926	.1606	.3737	.4600	.3906
BM25 + Bo1 + Clauses ($\alpha = 0.9$)	**.3957**	.1618	**.3772**	**.4625**	**.3975**

We can see that the proposed method improves results over the baseline and over query expansion, for all relevance measures including GMAP. This is very encouraging because it is one of the few results to our knowledge that show that WordNet information and WSD can be used to improve ad-hoc retrieval in an open domain.

Note that increasing values of α lead to increasing results[4]; however $\alpha = 1$ lead to poor results during the testing phase and was not submitted. This is somewhat surprising, reinforces the idea that one must be very conservative in query expansion in order to be robust. This requires further investigation. Further evidence of the robustness of our method is the fact that the use of AvICTF was not found useful.

In our opinion a bottleneck to further improve performance is in the creation of high quality structures. WordNet Similarity methods tend to produce noisy clauses, often putting in correspondence terms that are not related in the context of the query.

References

[1] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

[2] Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.*, 19(1):1–27, 2001.

[3] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[4] Ben He and Iadh Ounis. Query performance prediction. *Inf. Syst.*, 31(7):585–594, 2006.

[5] K. L. Kwok. Experiments with a component theory of probabilistic information retrieval based on single terms as document components. *ACM Trans. Inf. Syst.*, 8(4):363–386, 1990.

[6] C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceeddings of the 14th Text REtrieval Conference (TREC 2005)*, 2005.

[7] José R. Pérez-Agüera, Hugo Zaragoza, and Lourdes Araujo. Exploiting morphological query structure using genetic optimisation. In *NLDB*, pages 124–135, 2008.

[8] V. Plachouras, B. He, and I. Ounis. University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier. In *Proceeddings of the 13th Text REtrieval Conference (TREC 2004)*, 2004.

[4]Note that query inclusion requires that similarity is greater than α, so even for $\alpha = 0$ many terms are not assigned to clauses.

[9] S.E. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Thirteenth Conference on Information and Knowledge Management (CIKM)*, 2004.

[10] Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*, pages 232–241, 1994.

[11] J. J. Rocchio. Relevance feedback in information retrieval. In G Salton, editor, *The SMART retrieval system*, pages 313–323. Prentice Hall, 1971.

[12] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.