Interactive search refinement techniques for HARD tasks

Olga Vechtomova[*] Murat Karamuftuoglu[**] Eric Lam[***]

[*] Department of Management Sciences, University of Waterloo, Waterloo, Canada
ovechtom@engmail.uwaterloo.ca
[**]Department of Computer Engineering, Bilkent University, Ankara, Turkey
hmk@cs.bilkent.edu.tr
[***]Department of Computer Science, University of Waterloo, Waterloo, Canada
ekhlamlo@student.cs.uwaterloo.ca

Abstract

In our entry to the new HARD track, we have investigated two methods of interactively refining user search formulations. One method consists of asking the user to select a number of sentences that may represent relevant documents, and then using the documents, whose sentences were selected for query expansion. The second method is to show to the user a list of noun phrases, extracted from the initial document set, and then expanding the query with the terms from the phrases selected by the user. The results show that the second method is an effective means of interactive query expansion and yields significant performance improvements.

1. Introduction

The main goal of the HARD track this year is to explore what techniques could be used to improve search results by using two types of information: (1) extra-linguistic contextual information about the user and the information need, and (2) information dynamically provided by the user in response to topic clarification questions.

We decided to focus this year on designing techniques for dynamically eliciting additional topic-oriented search criteria from the users, and using their feedback in the second-stage search iterations. We submitted one baseline run, two clarification forms and two final runs. The following subsections describe the system design for each of the runs.

2. System description

2. 1 Baseline run

For all of our submitted runs we used Okapi BSS (Basic Search System). For the baseline run *UWAThard1* we used keywords only from the title fields of topics, as these proved to be most effective in our preliminary experiments described in section 2.2.3. The topic titles were parsed in Okapi, weighted and searched using BM25 function against the HARD track corpus. We used a GSL file, constructed on the basis of past TREC data, and comprised of 222 stopwords, 254 semi-stopwords (terms that are indexed, but not used in blind or relevance feedback), 58 phrases and 432 synonym groups.

2. 2 Query expansion method 1

According to the track specifications, a clarification form for each topic must fit into a 1152 x 900 screen and the user may spend no more than 3 minutes filling out each form. We have evaluated two clarification forms, the first of which – CF1 – is described in this section.

In brief, our first approach to eliciting user feedback was to take *N* top-ranked documents from the baseline run, select one sentence per document, include it in the clarification form CF1, and ask the user to select all sentences that possibly represent relevant documents.

The goal that we aim to achieve with the aid of the clarification form CF 1 is to have the users judge as many relevant documents as possible on the basis of one sentence per document. The main research questions that we explored here were:

RQ1: What is the error rate in selecting relevant documents on the basis of one sentence representation of its content?
RQ2: How does sentence-level relevance feedback affect retrieval performance?

2.2.1 Sentence selection

In more detail the sentence selection algorithm consists of the following steps:
From the baseline run, we take N top-ranked documents. Given the screen space restrictions, we could display only 15 three-line sentences, hence *N*=15. The full-text of each of the documents is split into sentences. For every sentence that contains one or more query terms, i.e. any term from the title field of the query topic, two scores are calculated: S1 and S2.

Sentence selection score 1 (S1) is the sum of *idf* of all query terms present in the sentence.

$$S1 = \sum idf_q \tag{1}$$

Sentence selection score 2 (S2):

$$S2 = \frac{\sum W_i}{f_s} \tag{2}$$

Where: W_i – Weight of the term *i*, see (3);
f_s – length normalisation factor for sentence *s*, see (4).

The weight of each term in the sentence, except stopwords, is calculated as follows:

$$W_i = idf_i(0.5 + (0.5 * \frac{tf_i}{t\max})) \tag{3}$$

Where: idf_i – inverse document frequency of term *i* in the corpus;
tf_i – frequency of term *i* in the document;
tmax – *tf* of the term with the highest frequency in the document.

To normalise the length of the sentence we introduced the sentence length normalisation factor *f*:

$$f_s = \frac{s\max}{slen_s} \tag{4}$$

Where: *smax* – the length of the longest sentence in the document, measured as a number of terms, excluding stopwords;
slen – the length of the current sentence.

All sentences in the document were ranked by S1 as the primary score and S2 as the secondary score. The rationale of our approach to sentence ranking is to pre-select, first, the sentences that contain more query terms, and therefore are more likely to be related to the user's query formulation, and secondly, from this pool of sentences to select the one which is more content-bearing and central to the topic of the document, i.e. which contains a higher proportion of terms with high tf*idf weights.

Next, since we are restricted by the screen space, we reject sentences that exceed 250 characters, i.e. three lines. In addition, to avoid displaying very short, and hence insufficiently informative sentences, we reject sentences with less than 6 non-stopwords. If the top-scoring sentence does not satisfy the length criteria, the next sentence in the ranked list is considered to represent the document.

Finally, since there are a number of almost identical documents in the corpus, we remove the representations of the duplicate documents from the clarification form using pattern matching, and process the necessary number of additional documents from the baseline run sets.

By selecting the sentence with the query terms and the highest proportion of high-weighted terms in the document, we are showing query term instances in their typical context in this document. Usually, but not always, a term is only used in one sense in the same document. Also, in many cases it is sufficient to establish the linguistic sense of a word by looking at its immediate neighbours in the same syntactic unit (i.e. a sentence or a clause). Based on this, we hypothesise that users will be able to filter out those sentences, where the query terms are used in an unrelated linguistic sense. We, however, recognise that it is more difficult, if not impossible, for users to reliably determine the relevance of the document on the basis of one sentence in those cases where the relevance of the document to the query is due to more subtle aspects of the topic, which are not evident from one sentence.

We evaluated CF1 on Financial Times and Los Angeles Times corpora from TREC volumes 4 and 5, and ad hoc topics 301-350. Forms were filled in by the authors. The average precision of user-selected sentences, calculated as the number of relevant sentences selected by the user out of the total number of sentences selected by the user, was 0.70 in our experiments. The average recall, calculated as the number of relevant sentences selected by the user out of the total number of relevant sentences shown in the clarification form, was 0.64. The average number of relevant documents shown was 5.36; average number of relevant selected sentences - 3.44; non-relevant selected – 1.44

2.2.2 Query expansion algorithm

The user's feedback to clarification form 1 is used for obtaining query expansion terms for the final run. Our approach to query expansion is to identify collocates of query terms – words co-occurring within a limited span with query terms. Previous query expansion experiments with long-span collocates of query terms obtained from 5 known relevant documents showed 72-74% improvement over the use of title only query terms on the Financial Times (TREC volume 4) corpus with TREC-5 ad hoc topics [4].

For the HARD track experiments we slightly modified our collocate extraction and selection algorithm. Instead of using fixed-size windows around instances of query terms to extract collocates, we define a window as one or more sentences surrounding the query term occurrence. The span of the window is measured as the number of sentences to the left and right of the sentence containing the instance of the query term. For example, span 0 means that only terms from the same sentence as the query term are considered as collocates, span 1 means that terms from 1 preceding and 1 following sentences are also considered as collocates.

In more detail the collocate extraction and ranking algorithm is as follows:
Each document judged relevant is split into sentences. For each query term we extract all sentences containing its instance, plus *s* sentences to the left and right of these sentences, where *s* is the span size. If *s*>0 we may have overlapping windows and extract the same sentence several times. To avoid this we keep the record of the sentences already extracted, so that each sentence is only extracted once.

After all required sentences are selected we extract stems from them, discarding stopwords. For each unique stem we calculate the Z score to measure the significance of its co-occurrence with the query term as follows:

$$Z = \frac{f_r(x,y) - \frac{f_c(y)}{N} f_r(x) v_x(R)}{\sqrt{\frac{f_c(y)}{N} f_r(x) v_x(R)}} \tag{5}$$

Where: $f_r(x,y)$ – frequency of x and y occurring in the same windows in the known relevant document set (R);
$f_c(y)$ – frequency of y in the corpus;
$f_r(x)$ – frequency of x in the relevant documents;
$v_x(R)$ – average size of windows around x in the known relevant document set (R);
N – corpus size in words.

The joint frequency of x and y – $f_r(x,y)$ is calculated as the product of $f(x)$ and $f(y)$ in that window.

All collocates with an insignificant degree of association: Z<1.65 are discarded, see [2]. The remaining collocates are sorted by their Z score.

After we obtain sorted lists of collocates of each query term, we select those collocates for query expansion, which co-occur significantly with two or more query terms. First, for each collocate the collocate score (C1) is calculated:

$$C1 = \sum n_i W_i \tag{6}$$

Where: n_i – rank of the collocate in the z-sorted collocation list for the query term *i;*
W_i – weight of the query term *i*.

The reason why we use the rank of the collocate in the above formula instead of its Z score is because Z scores of collocates of different terms are not comparable.
Finally, we rank collocates by two parameters:

(1) the number of query terms they co-occur with;
(2) C1 score.

Top *k* collocates in the ranked list are added to the original query terms.

2.2.3. Evaluation

We evaluated the algorithm with blind feedback, trying to find the optimal values for *R* - the size of the pseudo-relevant set, *s* – the span size, and *k* – the number of query expansion terms. The results indicate that variations of these parameters do not have significant effect on precision. However, some tendencies were observed, namely: (1) larger *R* values tend to lead to poorer performance in both Title and Title+Desc. runs; (2) larger span sizes also tend to degrade performance in both Title and Title+Desc runs.

The Title-only unexpanded run was 10% better than Title+Description. Similarly, the expansion of Title-only queries performed better than expansion of Title+Description queries. For example, AveP of the worst Title+Desc expansion run (R=50, s=4, k=40) is 23% worse than the baseline, and AveP of the best run (R=5, s=1, k=10) is 8% better than the baseline. AveP of the worst Title expansion run (R=50, s=5, k=20) is 4.5% worse than the baseline, and AveP of the best Title expansion run (R=5, s=1, k=40) is 10.9% better than the baseline.

Based on this data we decided to use terms only from the Title section of the topics for the official runs, and, given that values *k*=40 and s=1 contributed to a somewhat better performance, we used these values in all of our official expansion runs. The question of *R* value is obviously irrelevant here, as we used all documents selected by users in the clarification form.

2.3 Query expansion method 2

The second user feedback mechanism that we evaluated consists of automatically selecting noun phrases from the top-ranked documents retrieved in the baseline run, and asking the users to select all phrases that contain possibly useful query expansion terms. The research questions explored here were:

RQ3: Do noun phrases provide sufficient context for the user to select potentially useful terms for query expansion?

RQ4: How does query expansion based on user-selected phrases affect retrieval performance?

We take top 25 documents from the baseline run, and select 2 sentences per document using the algorithm described above. We then apply Brill's rule-based tagger [1] and BaseNP noun phrase chunker [3] to extract noun phrases from these sentences. The phrases are then parsed in Okapi to obtain their term weights, removing all stopwords and phrases consisting entirely of the original query terms. The remaining phrases are ranked by the sum of weights of their constituent terms. Top 78 phrases are then included in the clarification form for the user to select. This is the maximum number of phrases that could fit into the clarification form.

All user-selected phrases were split into single terms, which were then used to expand the original user query. The expanded query was then searched against the HARD track database using BM25 function for document retrieval and BM250 for passage retrieval.

3 Results

3.1 Document-level evaluation

The document-level results of the three official runs are presented in table 1. UWAThard1 is the baseline run using original query terms from the topic titles. UWAThard2 is an experimental run using query expansion method 1 plus the granularity and known relevant documents metadata. UWAThard3 is an experimental run using query expansion method 2 plus the granularity metadata.

Run	Run description	Soft relevance		Hard relevance	
		Precision @ 10	Average Precision	Precision @ 10	Average Precision
UWAThard1	Original title-only query terms; BM25 used for all topics	0.4875	0.3134	0.3875	0.2638
UWAThard2	Query expansion method 1; granularity and rel.docs metadata	0.5479	0.3150	0.4354	0.2978
UWAThard3	Query expansion method 2; granularity metadata	0.5958	0.3719	0.4854	0.3335

Table 1. Document-level evaluation results

UWAThard2 did not achieve statistically significant improvement over the baseline. This finding did not correspond to our initial test runs with the FT and LA collections, which showed 21% improvement over the original title-only query run. The analysis of the numbers of relevant and non-relevant sentences selected by the annotators showed slightly better results of 0.73 in average precision and 0.69 in average recall, compared to the selections that we made from clarification forms based on the FT and LA collections (see section 2.2.1). On average 7.14 relevant sentences were included in the forms. The annotators on average selected 4.9 relevant and 1.8 non-relevant sentences. Figure 1 shows the number of relevant/non-relevant selected sentences by topic. It is not clear why query expansion method 1 performed worse in the official UWAThard2 run compared to the test run on FT and LA collection, given very similar numbers of relevant sentences selected. Corpus differences could be one reason – HARD corpus contains a large proportion of governmental documents, and we have only evaluated our algorithm on newswire corpora. More experimentation is needed to determine the effect of the governmental documents on our query expansion algorithm.

In addition to clarification forms, we used the known relevant documents metadata for UWAThard2. We need to conduct more experiments without this metadata to determine how much it contributed to the performance of our query expansion method.

Our second query expansion run (UWAThard3) was among the best runs in the track, gaining an 18% improvement over the baseline in average precision in soft-doc evaluation and 26.4% in hard-doc evaluation, both of which are statistically significant (using t-test at .05 significance level). These are the highest improvements over the baseline among the top 50% highest-scoring baseline runs submitted to the track. Query expansion method 2 achieved lower performance gains in our training runs on FT and LA collections, which can be explained by the lower number of phrases selected. LDC annotators selected on average 19 phrases, whereas we selected on average 7 phrases in the training runs. This suggests that selecting more phrases leads to a notably better performance. The reason why we selected fewer phrases than the LDC annotators could be due to the fact that on many occasions we were not sufficiently familiar with the topic, and could not determine how an out-of-context phrase is related or not

related to the topic. The LDC annotators are more familiar with the topics, which they have formulated.

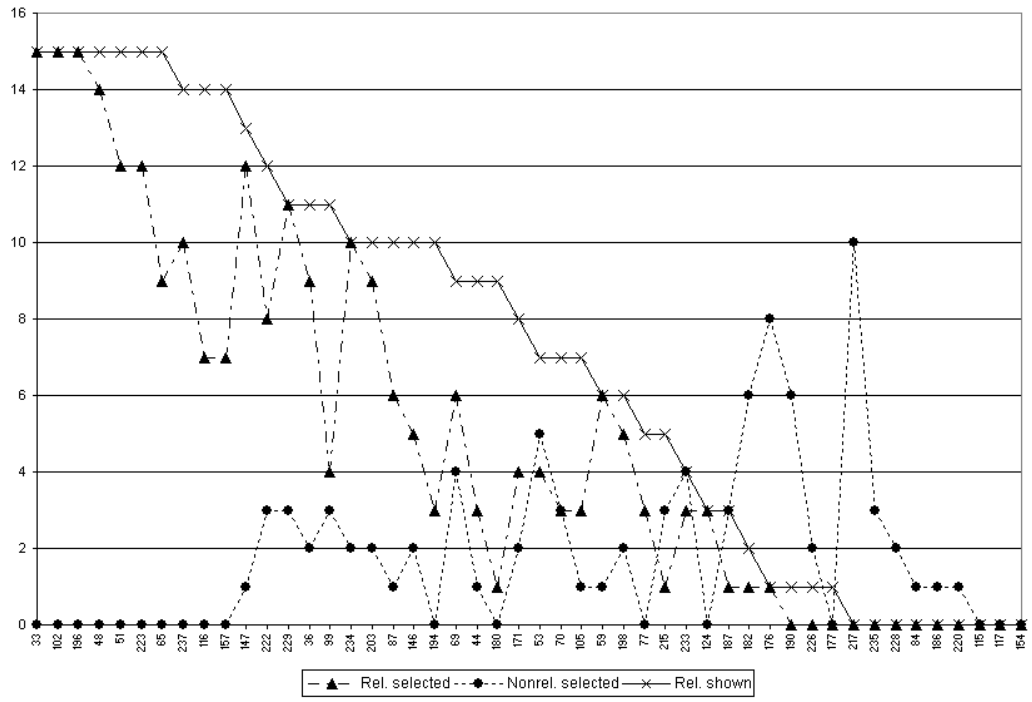

Figure 1. Sentences selected by LDC annotators from the clarification form 1.

3.2 Passage-level evaluation

Passage-level evaluation results are given in table 2. UWAThard3 showed 27% improvement in R-precision over UWAThard1, while UWAThard2 – 23%. Such big difference between the expansion runs and the baseline was expected, since we only did document-level retrieval for the baseline run. All our runs were above the median in all passage-level measures

Run	Passage P@10	R-Precision	F(30)
UWAThard1	0.2668	0.1908	0.1255
UWAThard2	0.3305	0.2359	0.1454
UWAThard3	0.3617	0.2426	0.1559

Table 2. Passage-level evaluation results

4 Conclusions and future work

This year we experimented with two user-assisted search refinement techniques:
(1) inviting the user to select from the clarification form a number of sentences that may represent relevant
documents, and then using the documents whose sentences were selected for query expansion.

(2) showing to the user a list of noun phrases, extracted from the initial document set, and then expanding the query with the terms from the user-selected phrases.

Comparison with other submissions to the HARD track (88 in total) shows that our submitted runs are above the median in all official evaluation measures. The second query expansion method is more promising than the first, and was among the best runs this year, achieving statistically significant improvement of 18% (soft-rel) and 26% (hard-rel) over the baseline.

In this year's entry we focused on utilising the user's feedback to clarification forms plus granularity and known relevant documents metadata. For the next year's entry, we plan to address other metadata: genre, familiarity and purpose.

Acknowledgements

This material is based on work supported in part by Natural Sciences and Engineering Research Council of Canada.

References

[1] Brill. E. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. Computational Linguistics, 21(4), 1995, pp. 543-565.

[2] Church K., Gale W., Hanks P., Hindle D. Using statistics in lexical analysis. *In Lexical Acquisition: Using On-line Resources to Build a Lexicon*, ed. U.Zernik, Englewood Cliffs, NJ: Lawrence Elbraum Associates, 1991, pp. 115-164.

[3] Ramshaw L., Marcus M., Text Chunking Using Transformation-Based Learning. Proceedings of the Third ACL Workshop on Very Large Corpora, MIT, June, 1995

[4] Vechtomova O., Robertson S.E., Jones S. Query expansion with long-span collocates. Information Retrieval, 6(2), 2003, pp. 251-273.