Microsoft Cambridge at TREC–12: HARD track

S E Robertson* H Zaragoza† M Taylor‡

1 Summary

We took part in the HARD track, with an active learning method to choose which document snippets to show the user for relevance feedback (compared to baseline feedback using snippets from the top-ranked documents). The active learning method is described, and some prior experiments with the Reuters collection are summarised. We also invited user feedback on phrases chosen from the top retrieved documents, and made some use of the 'relt' relevant texts provided as part of the metadata. Unfortunately, our results on the HARD task were not good: in most runs, feedback hurt performance, and the active learning feedback hurt more than the baseline feedback. The only runs that improved slightly on the no-feedback runs were a couple of baseline feedback runs.

2 Overview

The present team at Microsoft Cambridge may be regarded as the descendant of the Okapi team, working first from City University London and then from Microsoft. A summary of the contributions to TRECs 1–7 is presented in [4]. In these TRECs on various adhoc tasks we had concentrated on the weighting schemes and pseudo relevance feedback (blind feedback), and had developed the successful BM25 weighting function. However, we also took part in most of the early interactive tracks, and also developed iterative relevance feedback strategies for the routing task. Following up on the routing work, in TRECs 7–11 we took part principally in the adaptive filtering track (summarised in [6]). This work included developing alternative feature selection strategies, and also extensive analysis of thresholding; one outcome of the latter was a method of calibrating the BM25 score into an estimate of the probability of relevance.

For this year's TREC, we have entered only the HARD track. We have concentrated on the use of the clarification forms (one-shot interaction with the originator of the topic).

Since moving to Microsoft we have been working in part with a successor to Okapi, the Keenbow evaluation environment. The work reported in this paper was undertaken entirely with this new system, which is described in outline below.

*Microsoft Research Ltd, 7 J.J.Thomson Avenue, Cambridge CB3 0FB, UK, and City University, London, UK. email ser@microsoft.com

†Microsoft Research Ltd, 7 J.J.Thomson Avenue, Cambridge CB3 0FB, UK. email hugoz@microsoft.com

‡Microsoft Research Ltd, 7 J.J.Thomson Avenue, Cambridge CB3 0FB, UK. email mitaylor@microsoft.com

3 System

Keenbow is built in part using components from the MSSearch system, used in various Microsoft products including the SharePoint Portal Server. Although MSSearch maintains its own index of a traditional inverted file type, Keenbow can work with collection indexes stored as SQL tables; the distinction is largely a matter of performance (efficiency). That is, for large collections/indexes, it may be necessary for performance reasons at search time to use the native inverted file indexing system, while for smaller collections everything can be done within SQL. Clearly 'large' and 'small' are relative to the current hardware and low-level system state-of-the-art. In practice, all the experiments described here came into the 'small' category, and were run using Keenbow on a Microsoft SQL Server, running on an Intel Quad 700MHz Xeon with 3GB RAM.

The basic ranking algorithm in Keenbow is the usual Okapi BM25. The collection was preprocessed in a standard manner, using a 126 stop-word list and the Porter stemmer. In the context of query expansion (from relevance or blind feedback), feature selection is again based on usual Okapi methods – normally, the absolute term selection criterion described in [5]. As before, relevance feedback involves selecting a small number of terms from the known relevant documents, and weighting all selected terms (including the original topic terms) by the usual BM25 methods.

Currently Keenbow indexes predefined passages (we have not yet implemented in Keenbow the arbitrary window retrieval that we had in Okapi). For these experiments we defined passages at a level which comes somewhere in between paragraph and sentence – in other words, documents are broken into non-overlapping passages, each consisting of one or a few sentences.

4 HARD

The particular aspect of the HARD track which appealed to us was the opportunity to invoke a user-interaction phase actually involving the assessor who originated the topic. This is clearly highly artificial if we want to see it as a simulation of a genuine interactive system; however, it is the first time in TREC that we have had the opportunity to interact with the assessors, and it provides scope for some interesting experiments on what kinds of information might be elicited from users, and to what effect they might be put. We were less concerned with the metadata aspect of the track: we made minimal use of metadata.

An outline of the system is as follows. We put the original topic to the system in the usual fashion, and obtain the top-ranked retrieved documents. From these we select some to show to the user/assessor. The baseline system shows the top five documents, but the major experimental version shows five selected from the top 30 according to an active learning principle, as discussed below. What we show the user in each case is a short passage extracted from the document in a query-specific fashion: a query-specific snippet. In addition, we show the user some (max 15) 2-word phrases selected from the snippets according to a statistical measure, again described below. We invite the users to make 'relevance' judgements on each snippet and on each phrase (the form of the question is discussed below). The clarification form submitted to the user is made up out of these snippets and phrases.

On receipt of the completed clarification forms, we have made various runs using various parts of the returned information in different ways. We also make limited use of some of the metadata. Some of these runs were submitted as our official returns, and others have been evaluated since.

4.1 Basic methods

Okapi BM25 is used with the following parameters: $k_1 = 0.4$; $b = 0.75$; $k_3 = 0$ (the last means that duplicate query terms were ignored).

This procedure is used with the feedback obtained from the clarification forms, as discussed below. It is also used in the active learning stage, when we hypothesise various combinations of relevance judgements which the assessor might make on the documents presented.

Essentially the procedure is as described in many previous TREC reports and elsewhere. Each relevant document (or piece of text) is parsed to extract all terms as indexed. A table of statistics for the complete merged set of terms (including all original topic terms) is generated, a term selection value is calculated, and the top terms according to this value are selected for inclusion in the query. The various parameters for this process are as follows:

- Term selection function: Absolute function described in [5].
- Threshold for term selection: -8.
- Treatment of original topic terms: forced inclusion.
- Weighting after selection: original topic terms were given a boost in the expanded query by assuming that they occur in $rload$ out of $Rload$ mythical relevant documents, to be added to the r and R respectively concerning the actual relevant items. These parameters were set to $Rload = 20$ and $rload = 19$.

5 Active learning: document selection

In the usual *probabilistic learning* setting we are tying to estimate some function $f(x)$ from a collection of values $(x_i, f(x_i))$ (the training sample). However, in the *active learning* setting [3, 2] there is no pre-existing training sample, but rather we get to ask or *query* the function $f(x)$ with our chosen x values. In general, rather than querying points at random, it is much more advantageous to query points for which i) our incertitude is greatest and ii) obtaining an answer to our query will change our present model of the function the most. Active learning algorithms are used to choose these values in some optimal manner, exploiting properties of the function f to obtain the most information in the least number of points.

This problem is reminiscent of our problem in HARD. Here we wish to learn the probability of relevance of a document with respect to a query, $P(r = 1|d, q)$, were $r \in \{0, 1\}$ is a relevance indicator function, and d and q are the indexed documents. From a probabilistic learning perspective we would then need a collection of data points (q, d, r). We cannot do probabilistic learning as such yet, since we do not have any such data points[1], but perhaps we could *query* the judge for such values. We would need to present the judge with the query and a carefully selected batch of documents and ask him to reveal if the documents are or are not relevant. We would then use this information to i) update our approximation of the function of interest $P(r|d, q)$ and ii) select the next batch of points.

However, two things stand in the way of such an approach. The first problem is that active learning algorithms are tailored to each learning algorithm. Probabilistic active learning exploits properties of the learned function [3, 2]; in particular one needs to compute analytically how the introduction of a data point will change the approximation of the function. But in the case of Okapi feedback this is *a priori* unknown (as described above).

The second is that we only get a single chance to ask the judge! So iterative procedures are out of the question. All we can do is exploit the knowledge available in the query to form our initial approximation of $P(r = 1|d, q)$ and then select a batch of documents to be used as queries.

So in fact active learning will be used weakly, more as an inspiration than as a rigorous application of its principles. For this reason we call the resulting feedback algorithm *active feedback*.

5.1 Algorithm

We assume that we have the following:

- a query q,
- an indexed document collection $D := \{d_i\}_{i=1..N}$
- a retrieval function $\pi(d, q, F)$ which scores a document d for a given query q and a given *feedback set* of documents $F \subset D$.

[1] In fact the extra documents (metadata items with the tag *relt*) provided by the judges could be considered as such data-points, but unfortunately these were not available at form-generation time.

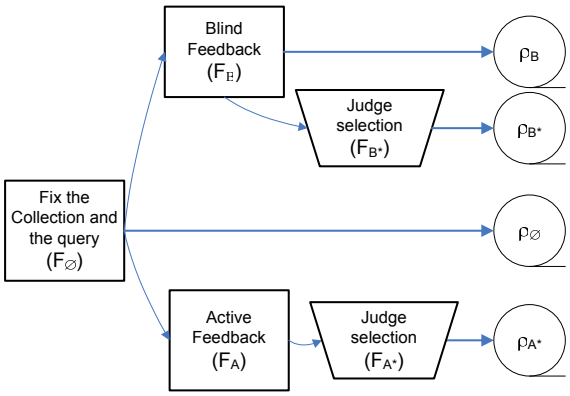

Figure 1: *Boxes and trapezes indicate automatic and manual procedures respectively. Feedback set names are indicated in parenthesis. Circles indicate the resulting collection rankings. Initially, a document collection and a query are fixed. The feedback set is empty and the resulting rank $\rho_\emptyset$ is the usual ad-hoc rank. By selecting first documents of this rank as the feedback set (F_B) we obtain the usual blind-feedback ranking ρ_B. If the judge is presented the documents in F_B she will select the relevant ones only (F_{B*}) obtaining an improved blind-feedback ranking ρ_{B*}. Finally, using the proposed active feedback procedure, we present the set F_A to the judge, who selects the relevant documents only (F_{A*}). This results in the ranking ρ_{A*}.*

We will assume that the function π models the probability of relevance of the document, given the query, the collection statistics and the feedback set F: $\pi(d,q,F) \approx P(r = 1|d,q,F,D)$. We do not assume any further knowledge about the retrieval function. In particular, we do not assume known the way in which the feedback set F modifies the query or the scoring function.

This *black-box approach* has the advantage of remaining quite general; in particular we will exploit the Okapi feedback framework which we know to be discontinuous with respect to the function parameters (and so difficult to analyze *inside the box*). However, the cost of this generality will be high: later, we will only obtain a heuristic algorithm instead of the usual provably optimal active learning algorithms.

Now we note that after fixing D, π and q, each feedback subset $F \subset D$ will implicitly define an *ordering* $\rho_F = \left(d_{(1)},...,d_{(N)}\right)$ of the documents on the collection, where $d_{(i)}$ is the document with rank i under $\pi(d,q,F)$.

Initially no human judgements are known and so $F = \emptyset$. This results in the baseline ad-hoc ranking of the collection, $\rho_\emptyset$. The usual *blind feedback* method sets F to the k highest scored documents. Let us denote this set $F_B := \left\{d_{(i)}|i \leq k\right\}$ for some value of k. The resulting collection ranking is denoted ρ_B (see Figure 1).

In the HARD setting, however, we can ask the user the relevance of a number of documents before defining the feedback set. Specifically, we could ask the user to verify the documents in the set F_B and select the relevant ones, forming the new set F_{B*}. This would result in the new (and hopefully improved) ranking ρ_{B*}.

We consider this "verified blind-feedback" method to be our HARD baseline. We believe we can improve on this selection because we feel that the top scoring documents, while they are the most *likely relevant* documents, are the *least informative* relevant documents, for two reasons: i) they will probably be very alike (the top documents are likely to be very redundant) and ii) the relevance of the documents is well explained by the query already.

Introducing the human judge as a filter has a crucial effect: we do not need to fear introducing irrelevant documents, since the human judge will eliminate them before retrieval! If we call F_A the set automatically selected for feedback, then the set used for feedback will be the human-verified set $F_{A*} \subseteq F_A$.

This allows us to look for more exotic documents that may be false but, *if they were relevant*, would carry a lot of new information on the query. Of course, since the judges have little time, we need to be slightly conservative or we risk not using any relevant documents (e.g. $F_{A*} = \emptyset$ if no relevant documents are selected in F_A) .

Therefore, we will argue that we need to detect not only the most *relevant* documents but also the most *informative* ones, or in other words, the ones that would produce the bigest change (or update) in the retrieval function if they were relevant. Unfortunately we do not have a good way to define the information gained by the introduction of a particular document in F. This is because we are considering a general retrieval function (a black-box), and therefore cannot analyse the effect of a relevant document in the function itself. All we can observe is the output of the black-box: the change induced in the ranking of the collection, ρ_F. For this reason we define the following function of the difference between two orderings:

$$\delta\left(\rho_A - \rho_B\right) := \frac{3}{\pi^2}\sum_{l=1}^{|\rho_A|}\left(\frac{1}{\rho_{l,A}} - \frac{1}{\rho_{l,B}}\right)$$

where $\rho_{l,A}$ indicates the lth coordinate of the vector ρ_A, and the constant $3/\pi^2$ is a normalisation factor which keeps δ in the [0,1] range. This function is chosen on the basis of the following criteria:

- it should have the value 0 if the two rankings are identical, and approach 1 if they are very different;
- it should depend more on the top end of the ranking than on items further down.

This second point is achieved by using the reciprocal ranks instead of the ranks themselves (in the same way that known-item search tasks are often evaluated using mean reciprocal rank). Note that it does *not* behave like the inverse of a rank correlation coefficient, specifically in that it has no notion of a reverse correlation.

We can finally state our objective: we need to choose the set F_{A*} (of some fixed size k) that maximises the quantity $\delta\left(\rho_\emptyset, \rho_{A*}\right)$.

Unfortunately, we do not know which documents will be chosen by the judge, and so we do not have access to F_{A*}. So we will revert to maximising the *expected* change of rank over all possible judge selections (weighted by the probability that these selections are relevant under π). For this, let us denote by $\mathcal{F}$ the power set[2] of F and by $\mathcal{F}_k$ the set of all subsets of F with size k or less including $\emptyset$. With this, we can define the *expectation* of δ over the set of documents F under $\pi(d, q, \emptyset)$ as:

$$\mathsf{E}\left[\delta_F\right] := \sum_{F' \in \mathcal{F}} \left[\delta\left(\rho_\emptyset, \rho_{F'}\right) \prod_{d_i \in F'} \pi\left(d_i, q, \emptyset\right) \right]$$

Finally, we define the *active feedback set* F_A as the subset of the collection D of some fixed size k which maximises $\mathsf{E}\left[\delta_{F_A}\right]$:

$$F_A = \arg\max_{F \in \mathcal{D}_k} \mathsf{E}\left[\delta_F\right]$$

In practice the size of $\mathcal{D}$ is too large to exactly compute 5.1. But we notice that for most documents their probability of relevance is so low that they would bring to zero any expectation in which they are considered as candidates. Therefore it is safe to consider only the most relevant documents as candidates. We do this simply by considering only the documents in D with highest $\pi(d, q, \emptyset)$ values. For our the HARD 2003 runs we considered only the top 30 documents. The calculation of $\delta\left(\rho_A - \rho_B\right)$ is based on comparing the rankings of the top 500 documents (as indicated, it is most strongly affected by changes at the top of the ranked list).

6 Phrase selection

The two-word phrases to be shown to the user in the clarification forms were selected as follows:

We considered each pair of adjacent words in every snippet shown to the user. For each such pair, we calculated the following *plausibility* measure (originally used in [1]): If s and t are two terms with frequencies $n(s)$ and $n(t)$ respectively the plausibility of the adjacent pair st is $n(st) \times C/(n(s)n(t))$, where C is the total number of tokens in the collection. For randomly collocated terms we would expect this measure to be around 1; we set a high threshold on it to select words which are collocated considerably more often than that. The selection threshold chosen was 20. We also chose phrases with a reasonable frequency of occurrence ($n(st) > 10$). Finally, we calculated the offer weight or term selection value, on the blind feedback assumption that the snippets chosen are all relevant (this is of course before we have user judgements). Thus the complete criterion was:

- Select phrases with plausibility> 20;
- From these, select those with $n(st) > 10$;
- Sort these by term selection value;
- Accept the top 15, or those with term selection value> 3, whichever is the less.

The resulting phrases mostly looked like reasonable phrases; some not. An example list from Topic HARD-033 is: *antimicrobial drug; APHIS regulations; hog cholera; intestinal tract; contagious disease; Endangered Wildlife; Nacional de; golden eagle; occurring outside; animal drugs; drug resistant; animal product; Shanxi Province; draft guidance; wild animals.*

7 Clarification forms

7.1 Retrieved items

As indicated, our principal aim was to obtain relevance feedback data from the assessors. However, given the various limitations (screen real estate and time taken to complete) on the clarification forms, it was not feasible to present the assessors with anything like complete documents. In a reasonable compromise between document numbers and amount of information per document, we decided to present up to four lines from each of up to five documents.

At the time of indexing, each document is partitioned into predefined, non-overlapping passages. Each passage is a single sentence or a small number of contiguous sentences. We therefore presented the best-matching passage from each of the selected documents in the form. In cases where the selected passage was too long, it was arbitrarily truncated. Most passages presented would include at least some of the query terms, but some would not, because of this arbitrary truncation. We considered including the complete passage in a small scrollable window in these cases, but rejected this idea, both for technical reasons (the version of Netscape being used by the assessors) and because it seemed counter to the principle of a restricted clarification form.

The issue of what question to ask the assessors about each document was an interesting one. Perhaps unlike many users of IR systems, they can be expected to have a rather clear idea about what 'relevant' might mean, given that they either have already made, or will in the near future be making, official TREC relevance judgements. On the other hand, the official judgements they will be making will be on the basis of reading (or at least being able to read) the entire document being judged. It seems a little hard to ask them to make an equivalent judgement on the basis of the snippet presented.

One of our interests is in the use of indirect evidence such as click-through as a form of feedback. We therefore decided to present the relevance question to the assessors as a click-through question:

[2] that is the set of all subsets of F including $\emptyset$. It is usually noted F^* but this clashes with our notation.

Assume that you have issued a query on the above topic to your search engine, which has responded with the following list. Would you click through to any of these documents? Check as many or as few as you like. * If you can answer your question from the snippet alone, please check "No need".

The radio buttons beside each item were:

- Yes
- Perhaps
- No
- No need *

The default button was 'No'. The 'Perhaps' was included primarily for the comfort of the assessors who might find it difficult to make a definite answer in some cases, but allows us to try the relevance feedback with or without the *Perhaps* responses included as relevant. The 'No need' button was included on the basis that some of the questions could be answered with a sentence or phrase which might actually be in the snippet. These were counted as relevant (although in such cases relevance feedback seems a bit superfluous).

The responses were coded 3 (No need), 2 (Yes), 1 (Perhaps), 0 (No) for the experiments discussed below.

7.2 Phrases

Phrases were selected from the snippets chosen for the documents shown to the assessor (but before truncation). Up to 15 were selected. The question asked was:

Do any of the following phrases help to describe what you are looking for? Check as many or as few as you like. * If you think a phrase is indicative of a document you do not want to see, please check "Neg".

The radio buttons for each phrase were:

- Yes
- No
- Neg*

The default button was 'No'. The 'Neg' (negative) button was included on the grounds that 'No' was neutral (*No* phrases would simply be ignored), but some phrases seem to indicate an incorrect context, and might therefore be treated in a more strongly negative fashion, as providing positive evidence *against* the relevance of the document. This was quite a popular button among the assessors, but raises interesting questions of how the negative evidence should be used, discussed further below.

These responses were coded 1 (Yes), 0 (No), -1 (Neg) for the experiments discussed below.

There was no necessary reason to choose the phrases from the chosen snippets – we could have chosen them from the (whole) chosen documents, or from some other set of documents. The data we have collected from the experiment allows us to simulate two more possibilities, by using the phrases selected for the baseline run with the snippets selected for the main experimental run, and vice versa.

8 Use of feedback data and metadata

When we have received the assessors' responses to the clarification forms, we have various forms of data that might be used in various ways and in various combinations in feedback. We have tried a few of these combinations as officially submitted runs, and some additional combinations are also evaluated in this paper.

8.1 Evaluated snippets and *relt* items

Snippets evaluated as relevant (in the click-through sense) are to be used for relevance feedback. In common with most other relevance feedback experiments, we make no use of items judged not relevant – they are simply ignored (instead, statistics from the whole collection, excluding those documents known to be relevant, are taken to represent the non-relevance class). Furthermore, we use the items judged relevant only in the usual relevance feedback algorithm: although it is likely that these items rise in the ranking as a result of the feedback, there is no necessary reason why they should rise to the top, and we do not force them to do so.

In the present circumstances, there is a choice between taking as the texts of the relevant items just the snippets judged relevant by the assessors, or the entire documents from which they come. We have chosen to take just the snippets themselves, on the grounds that those are the items of text actually judged (but in the cases where the snippet was truncated for display, we take the entire snippet). It may be argued that this approach does not fit very well with the theory on which the relevance feedback algorithm is based, which involves counting documents containing each term. This is an issue for further work.

One of the metadata items to which we now have access is the 'relt' item – that is, any texts provided as relevant by the assessor in advance of the search. One issue associated with these *relt* items, interacting with the issue just mentioned, is their length – they are typically quite long, certainly much longer than our snippets, and probably comparable in length to the documents. In the experiments where we have included the *relt* items, we have treated them in the same way as the relevant snippets. However, it seems likely that some differentiation should be made.

8.2 Positive phrases

It would be possible to treat any phrase as if it were a (new) single term, and give it a weight on the same basis that a term would be weighted. However, this ignores the fact that the phrase may contain terms that are themselves in the query. In this case, the danger is that a document will be overweighted because it gets the weight of the phrase and also the weight of the single term contained in the phrase. To put it another way, the probabilistic model makes independence assumptions, but in this case we have an extreme dependence situation: the pres-

ence of the phrase implies the presence of any constituent single term.

Since the constituent terms may or may not be in the query, we have a set of cases to deal with. Also, a phrase has a 'natural' weight of its own (the usual RSJ weight which is the document-independent part of the BM25 formula, which reduces to a tf*idf weight in the absence of relevance information but is a relevance weight when we have such information). This 'natural' weight may or may not exceed the combined weights of the constituent terms.

Thus our algorithm looks like this. We consider only 2-term phrases ab, and w(x) is the natural weight of x, which can be single term or phrase. wPhrase will be the weight to be given to the phrase.

wPhrase $\leftarrow$ w(ab)
IF (a $\in$ query) THEN wPhrase $\leftarrow$ (wPhrase - w(a)) ENDIF
IF (b $\in$ query) THEN wPhrase $\leftarrow$ (wPhrase - w(b)) ENDIF
IF (wPhrase $<$ 0) THEN wPhrase $\leftarrow$ 0 ENDIF

8.3 Negative phrases

Negative phrases present some of the same problems as positive ones – namely, any of the constituent terms may or may not be in the query. In addition, there is another general problem about using negative weights. The probabilistic theory that is the basis for BM25 is quite at home with negatively-weighted terms – essentially any term whose presence in a document is evidence against relevance – but for several practical reasons, negative weights have been avoided in almost all work with BM25. The normalisation of BM25 is designed to ensure that an absent term contributes nothing to a document's score, which means that documents containing none of the query terms (usually the vast majority of documents) have zero score. This is a big advantage in a system based on inverted files. Furthermore, if the query contains only positively weighted terms, then this large set of zero-scored documents is necessarily at the bottom of the ranking. Thus a ranking of all the non-zero (and therefore positive) scores implies in a very straightforward way a ranking of the complete collection (and of course no user ever ventures into the large mass of zero-scored documents tied at bottom rank). The usual term selection algorithms that form part of relevance feedback tend to select only positively weighted terms.

Introducing negative term weights potentially complicates this picture. In practice, however, small negative weights for a small number of terms may be accommodated (we would presumably only ever look at documents with resulting positive score, and ignore not only the zeros but also the net negatively scored documents).

In the light of these considerations, the proposed treatment of *Neg* phrases is as follows. The principle is that if either (or both) of the constituent terms is in the query, occurrences of that term in the document *as a constituent of the phrase* should be ignored (that is, should not contribute to the s, but other occurrences of the term on its own should continue to count positively. There is a slightly complex interaction here with the tf factor which is the other bit of BM25, and the proposed algorithm does not deal very elegantly with this interaction, but may serve as a first approximation. In addition, the presence of the phrase in a document should somewhat reduce the score of the document. The 'natural' (quite likely positive) weight of the phrase does not figure in this algorithm; however, we begin by assigning the basic amount by which the phrase should reduce the score. This might be a small positive constant, or perhaps half the average weight of the single query terms, or the weight of the least-weighted single query term. Then we consider the cases.

define small wDown $>$ 0
wPhrase $\leftarrow$ - wDown
IF a $\in$ query THEN wPhrase $\leftarrow$ (wPhrase - w(a)) ENDIF
IF b $\in$ query THEN wPhrase $\leftarrow$ (wPhrase - w(b)) ENDIF

There is clearly scope for many experiments here. In the event, because of the generally negative results from the other experiments discussed (and our efforts to understand them), we have not yet conducted any experiments on these negative phrases.

8.4 Topic description and metadata

As a guiding principle, we tried to limit the amount of information required *a priori* form the user. To this end, we used only the *Title* of the topic description (discarding the topic's description and narrative) and discarded most of the topic's metadata. The two exceptions were:

GRANULARITY If the value was SENTENCE or PHRASE, we returned the best-matching passage as the passage-definition in the retrieved document (after ranking the documents by the usual document score).

RELATED-TEXT We used these texts in the same way that we used fragments returned in the clarification forms as relevant (see experiments below).

9 Experiments

9.1 Preliminary experiments

Before deciding on the methods to be used for HARD, we made a series of runs based on the active learning idea with the Reuters RCV1 corpus (as used in recent years for the adaptive filtering track), with the topics generated for last year's filtering track. We did not have the possibility of interaction with the assessors in this case, so the experiments simulated user feedback (or rather an upper bound) by assuming that the user would recognise as relevant the chosen snippet from a document that was officially judged as relevant.

In other respects these experiments were similar to those conducted for HARD – that is, for the active learning proce-

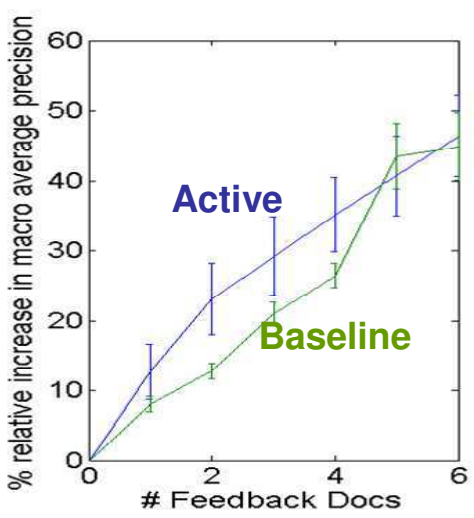

Figure 2: *Results on Reuters RCV1 corpus, TREC 2002 filtering topics. Performance is shown after feedback on 1–6 documents. In the case of the Baseline, these are the top-ranked documents; in the case of active learning, they are those selected from the top 30 by the active learning algorithm.*

dure we chose the best snippets according to the above algorithm, without reference to relevance judgements. Having chosen the snippets, we looked up their relevance judgements in lieu of actually consulting a user, and used the snippets from relevant documents to expand the query. The baseline here was to choose the top-ranked documents to provide the snippets and the relevance judgements.

These experiments and results are not described in detail here, but Figure 2 shows some results. Both active learning and baseline feedback improve on the baseline without feedback. On the whole the active learning procedure does better than the baseline feedback with few judged documents; this advantage may have disappeared by the time five documents have been used for feedback. Nevertheless, the results from these experiments were sufficiently encouraging for us to adopt the active learning method in our HARD experiments.

9.2 Variables and runs

The initial run, submitted before the clarification forms, is called MSRCbase. This is a straight BM25 baseline run on the topic titles only, and was the basis for the construction of the clarification forms.

[Actually, we believe that the run we submitted as baseline run was not the correct one. The submitted run was somewhat better than the 'real' baseline. The results reported below include the correct baseline run. They do not, however, change the generally negative results of this paper.]

As indicated above, we submitted two sets of clarification forms, one based on snippets from the top 5 ranked documents on the baseline run, and the other on the items selected by the active feedback analysis described above. (However, we attempted to remove duplicates from the baseline run snippets, after selection of the top 5, so that we often presented less than 5 snippets. The active feedback algorithm could be expected to remove duplicates anyway.)

Thus we had the following main variables to experiment on:

- use of the snippets in relevance feedback;
- use of the relt texts from the metadata in a similar fashion;
- and use of the phrases.

Our official runs were coded MSRCsXeXpX and MSRCsXeXpXB where the Xs are defined below and the B indicates use of the baseline clarification forms (rather than the Active Feedback ones). The use of snippets is coded s1, s2 or s9 – s9 means no snippets were used in feedback, s2 means that only the 'Yes' and 'No need' snippets were used (referred to below as *best* snippets), and s1 means that the 'Perhaps' snippets were also used (referred to as *good* snippets). e1 means the extended (relt) texts from the metadata were used, e0 that they were not. p1 indicates that the positive phrases were used, p0 that they were not (the negative phrases were not used in the official runs). We submitted these runs:

Run	CFs	Snippets	relt texts	phrases
MSRCs1e1p1	AF	good	yes	positive
MSRCs1e0p1	AF	good	no	positive
MSRCs1e0p0	AF	good	no	no
MSRCs9e1p1	AF	none	yes	positive
MSRCs2e0p1	AF	best	no	positive
MSRCs9e1p0	none	none	yes	no
MSRCs1e1p1B	base	good	yes	positive
MSRCs1e1p0B	base	good	yes	positive
MSRCs1e0p0B	base	good	yes	positive

We have since completed additional runs with other combinations of these variables.

9.3 Results

Unfortunately, our results have been almost exclusively negative. That is, we failed to improve significantly on the baseline with any of our methods; most of them degraded performance. Furthermore the active learning methods degraded performance more than the baseline feedback runs. The main results are in Table 1. The only run that outperforms the baseline uses the top 5 best snippets only, no phrases or relt texts.

We wished to test the hypothesis that the difference from our earlier Reuters experiments had to do with the fact that we used official relevance judgements in the Reuters experiments. We therefore made some runs on the HARD topics based on the selected snippets, but looking up official relevance judgements rather than using the feedback provided to the clarification forms. However, although this gave slightly better performance than our official runs, we still do not get anything like the increases observed in the Reuters experiments (see Table 2).

Table 1: Main results

Run	MAP	P@10	Notes
[MSRCbase]	.285	.496	Our corrected version, not as submitted
MSRCs1e0p0	.239	.467	Feedback from active learning snippets
MSRCs1e0p1	.215	.421	– plus phrases
MSRCs1e1p1	.255	.488	– plus relt texts
MSRCs1e1p0*	.251	.454	– relt texts but no phrases
MSRCs1e0p0B	.282	.490	Feedback from top 5 snippets
MSRCs1e0p1B*	.251	.446	– plus phrases
MSRCs1e1p1B	.277	.492	– plus relt texts
MSRCs1e1p0B	.291	.494	– relt texts but no phrases
MSRCs2e0p0*	.259	.488	Active learning best snippets only
MSRCs2e0p0B*	.297	.504	Top 5 best snippets only
MSRCs9e1p0	.251	.452	Relt texts only, no feedback
Note: The results here differ slightly from the official ones. This is probably due to a small difference in our method of calculation of the measures from trec_eval. We will be attempting to locate and remove this difference.			
Note 2: Runs marked * are additional to the official runs.			

Table 2: Feedback using official relevance judgements

Run	MAP	P@10	Notes
MSRCs1e0p0-R	.265	.488	Active learning snippets, official rels
MSRCs1e0p0B-R	.273	.485	Top 5 snippets, official rels

10 Conclusions

We are obviously disappointed at the results obtained. They suggest that our basic feedback methods are fragile with regard to some or all of the following: the collection, the nature of the documents, the use of snippets for feedback, the topics. . . Given that feedback on the top five documents (baseline feedback) hurts us, it is perhaps not surprising that active learning feedback hurts us more. We have some serious work to do!

References

[1] M M Beaulieu et al. Okapi at TREC–5. In E M Voorhees and D K Harman, editors, *The Fifth Text REtrieval Conference (TREC–5)*, pages 143–165. Gaithersburg, MD: NIST, 1997. NIST Special Publication 500-238.

[2] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 705–712. The MIT Press, 1995.

[3] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[4] S E Robertson and S Walker. Okapi/Keenbow at TREC–8. In E M Voorhees and D K Harman, editors, *The Eighth Text REtrieval Conference (TREC–8)*, pages 151–162. Gaithersburg, MD: NIST, 2000. NIST Special Publication 500-246.

[5] S E Robertson and S Walker. Microsoft Cambridge at TREC-9: Filtering track. In E M Voorhees and D K Harman, editors, *The Ninth Text REtrieval Conference (TREC–9)*, pages 361–368. Gaithersburg, MD: NIST, 2001. NIST Special Publication 500-249.

[6] S E Robertson, S Walker, H Zaragoza, and R Herbrich. Microsoft Cambridge at TREC 2002: Filtering track. In E M Voorhees and D K Harman, editors, *The Eleventh Text REtrieval Conference, TREC 2002*, pages 439–446. Gaithersburg, MD: NIST, 2003. NIST Special Publication 500-251.