Microsoft Cambridge at TREC–13: Web and HARD tracks

Hugo Zaragoza* Nick Craswell † Michael Taylor ‡ Suchi Saria § Stephen Robertson¶

1 Overview

All our submissions from the Microsoft Research Cambridge (MSRC) team this year continue to explore issues in IR from a perspective very close to that of the original Okapi team, working first at City University of London, and then at MSRC.

A summary of the contributions by the team, from TRECs 1 to 7 is presented in [3]. In this work, weighting schemes for ad-hoc retrieval were developed, inspired by a probabilistic interpretation of relevance; this lead, for instance, to the successful BM25 weighting function. These weighting schemes were extended to deal with pseudo relevance feedback (blind feedback). Furthermore, the Okapi team participated in most of the early interactive tracks, and also developed iterative relevance feedback strategies for the routing task.

Following up on the routing work, TRECs 7–11 submissions dealt principally with the adaptive filtering task; this work is summarised in [5]. Last year MSRC entered only the HARD track, concentrating on the use of the clarification forms [6]. We hoped to make use of the query expansion methods developed for filtering in the context of feedback on snippets in the clarification forms. However, our methods were not very successful.

In this year's TREC we took part in the HARD and WEB tracks. In HARD, we tried some variations on the process of feature selection for query expansion. On the WEB track, we investigated the combination of information from different content fields and from link-based features.

Section 3 briefly describes the system we used. Section 4 describes our HARD participation and Section 5 our TREC participation.

2 System

The system is the Keenbow experimental environment as described in [6]. The experiments described here 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 collections were preprocessed in a standard manner, using a 126 stopword list and the Porter stemmer (where stemming is used).

3 HARD Track

For the experiments we submitted to this year's HARD track, we concentrated on methods for query expansion to improve relevance feedback. More formally, our experiments addressed the following problem: given a fixed ranking function $\mathcal{F}$, a query Q, ranked list of documents $\mathcal{D}_Q$ ranked with respect to Q, and few $(0-5)$ snippets corresponding to documents in $\mathcal{D}_Q$ marked by the user as relevant to the query, how can we use the relevant snippets to improve the ranked list $\mathcal{D}_Q$?

3.1 Feature selection methods

We are interested in two problems: i) selecting new terms to be added to the query, and ii) weighting these terms. For the selection problem, we investigate the use of several functions. We call these functions *feature selection* functions (noted $\sigma(t_i)$). For the weighting problem we are using standard RSJ feedback weights, except that terms in the query are given artificially high R and r counts.

We tried two types of feature selection functions: *relative* and *absolute*. Relative feature selection measures produce an ordering on the candidate terms, but they do not give us an indication on the number of terms to be included. The magnitude of the value $\sigma(t_i)$ depends on the query in an unknown way. We use relative functions by deciding on a fixed number of expansion terms for all queries. Robertson Selection Value is the only relative function we used.

Absolute feature selection values, on the other hand, can be tested against a threshold (query-independent) to decide how many terms to be included. Typically, queries with a large number of relevant documents can select more terms for inclusion.

Before discussing the different feature selection functions we used, we introduce some useful notation:

- N is the total number of documents in the corpus.
- $|V|$ is the number of distinct terms in the corpus.
- R is the total number of relevant documents (for a fixed query).

*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 nickcr@microsoft.com

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

§Stanford Universit, Dept. Comp. Sci., 353, Serra Mall, Stanford, CA 94305. email ssaria@stanford.edu

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

- r_i is the number of relevant documents in which term t_i appears.
- n_i is the total number of documents in which term t_i appears.
- Probability of term t_i in relevant class: $p_i = \frac{r_i}{R}$.
- Probability of term t_i in non-relevant class: $\bar{p}_i = \frac{n-r_i}{N-R}$.
- Probability of term t_i in corpus: $p'_i = \frac{n_i}{N}$.
- $C^R_{r_i} = \frac{R!}{r_i!(R-r_i)!}$ is the number of combinations of r_i in R.

We experimented with the following feature selection functions:

- Robertson Selection Value (RSV) [2]

$$\sigma(t_i) = (p_i - \bar{p}_i)log(\frac{p_i(1-\bar{p}_i)}{(1-p_i)\bar{p}_i}) \approx r_i log(\frac{p_i(1-\bar{p}_i)}{(1-p_i)\bar{p}_i})$$

- Significance Rule (SGN) [4]

$$\sigma(t_i) = r_i log\frac{N}{n_i} - log(C^R_{r_i}) - log|V|$$

- Maximum Likelihood Decision Rule (MLDR)

$$\sigma(t_i) = r_i \log(\frac{p_i(1-\bar{p}_i)}{(1-p_i)\bar{p}_i}) + R\log(1-p_i) + n_i \log\frac{\bar{p}_i}{p'_i}$$

- Simplified Kullback-Leibler Distance (KLD) [1]

$$\sigma(t_i) = p_i \log\frac{p_i}{\bar{p}_i}$$

- Modified Chi-Squared (CHI2) [1]

$$\sigma(t_i) = \frac{(p_i+\bar{p}_i)^2}{\bar{p}_i}$$

The MLDR derives from a maximum likelihood argument, which will be explored in a later paper. The KLD is a simplification of the Kullback-Leibler distance between two probability distributions. In this case the distributions relate to the events of (presence, absence) of a given term. The full KL distance would be

$$p_i \log\frac{p_i}{\bar{p}_i} + (1-p_i)\log\frac{1-p_i}{1-\bar{p}_i}$$

but the second term is ignored here, as is the fact that the KL distance is asymmetric between the two distributions. The simplification also has the effect that terms will not be selected because they are good indicators of *non-relevance*.

3.2 Experiments on Feature Selection

To test for the performance of the feature selection functions for query expansion, we made an experimental setup similar to the HARD'04 test scenario. We generated clarification forms with snippets and first passages of the top five retrieved documents for each query in the HARD'04 training set. We marked the snippets as relevant only if the corresponding document in the corpus had a positive relevance judgment. Using this, we extracted only queries $(304, 306, 307, 313, 315, 317, 320, 326, 327, 328)$ for which we got one or more relevant snippets and used this as the set to test for query expansion using different feature selection functions. Figure 1 shows the plot for mean average precision against the number of words in the expanded query for these 10 topics.

It may be noted that all selection value formulae produce more-or-less comparable peaks. Although two other measures show slightly higher peaks, KL distance (KLD) has least variation in the mean average precision for the different number of words added; in other words, it seems to be less susceptible than the others to non-optimal setting of the threshold. We tested these measures as well on the Reuters Vol.1 corpus and HARD'03 obtaining similar results. For example, DLF consistently gave high mean average precision on query expansion using (1-3) documents for queries in the Reuters collection. Hence, we chose to use KLD for query expansion on our submitted runs to HARD04.

Intuitively, KL distance values the dissimilarity between the distribution of the term in the relevant class and the entire corpus. In other words, words that are infrequent in the corpus but frequently present in the relevant documents are considered most informative of relevance and hence best candidate terms for inclusion to the query. However, as with all the absolute methods, for a given score threshold there is large variance in the number of words selected for different queries (1-30). In particular, we found that it often over-estimates the number of terms to be included in the query. For this reason, we introduced a second parameter which limits the maximum number of words to be included in a query (noted maxT).

Figure 2 shows the mean average precision over queries when using the KLD selection function against an absolute threshold (plotted in the x-axis), for different values of maxT. The 'infinity' line corresponds to the basic KLD measure without maxT threshold. Note that the use of maxT is beneficial and obtains te best average precision (on the training set at least).

3.3 Metadata and clarification forms

The clarification forms contained three forms of data: Snippets, Good Phrases and Bad Phrases. That is, we showed the user a snippet from each of the five top-ranked documents from the baseline search (passage retrieval methods were used to identify appropriate snippets, but we made no further passage retrieval runs, and did not make submissions for the passage

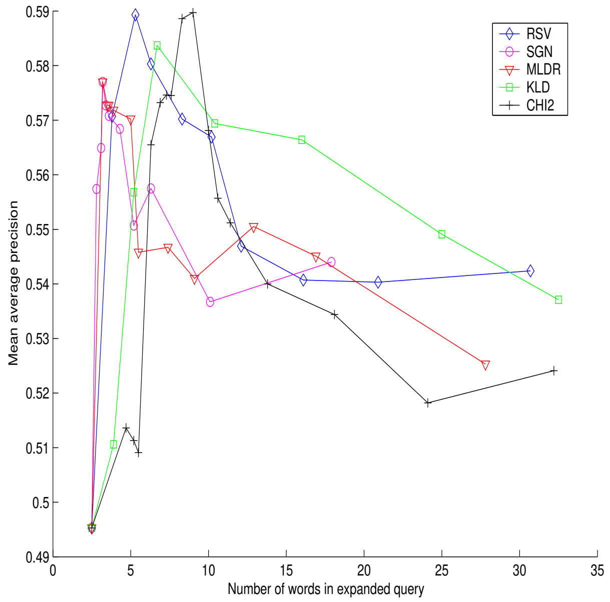

Figure 1: Mean average precision variation as selected words are added to the query. The baseline result of search using only the original query is shown as the first data point with number of words = 2.5

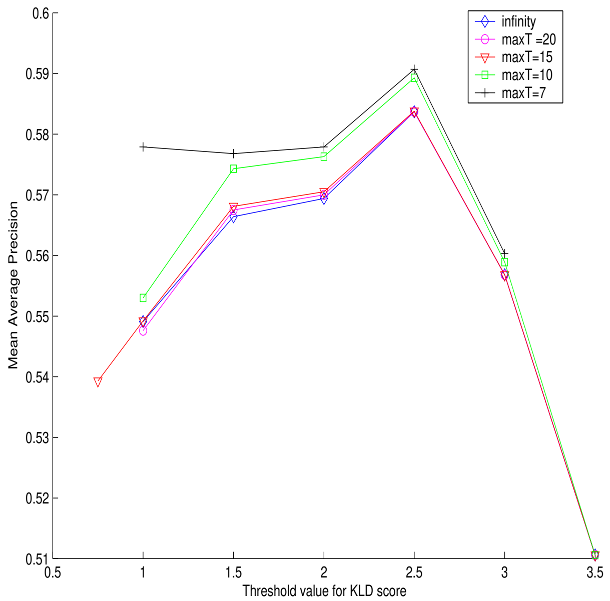

Figure 2: Mean average precision variation as the KLD threshold is varied, for different number of maximum terms (maxT).

retrieval evaluation). Users were asked whether they would click on each snippet; responses 'Yes' or 'Perhaps' or 'No need' were taken as relevant (the last category means that the user could infer the answer they wanted directly from the snippet, without going to the original document). In addition, they were invited to add up to five words or phrases indicating Good documents, and up to five indicating Bad documents.

We used these in various ways and in various combinations for feedback. We made minimal use of the metadata as we were primarily interested in assessing relevance feedback. The only form of metadata we used besides the query title, was the 'Description' field. Often, the query title did not convey sufficient information on which the relevance judgements were made. For example, the title for Topic 422 was "Video Game Crash" but the description in the data was "Is the market for interactive software just waiting to crash?" Since our original search was based only on the query title, our results presented on the clarifications forms discussed video game crashes instead of market crash for video game software and hence were all marked as non-relevant. We used the description meta-data along with the query to generate results for such queries in the final run.

More specifically, metadata and feedback data were used as follows in the submitted runs: If *good phrases* were used, they were added to the query. If *bad phrases* were used, they were removed from the query. If *snippets* were used, all the terms of the snippets were considered for inclusion using the feature selection algorithm described bellow. Finally, if *description*s were used and there were no relevant snippets and there were no good phrases, all the terms in the description were added to the query.

Terms considered for inclusion (from relevant snippets) were selected by the following procedure (as described in the previous section):

PROCEDURE FeatureSelectKLD
(
  S = set of all words in relevant snippets ,
  ST = score threshold ,
  MaxT = maximum number of terms to be included ,
)
{
  newSet = $\emptyset$
  FOREACH($t_i \in S$) {
    IF ($t_i >$ ST)
      { newSet = newSet $\bigcup$(t_i,SKLD(t_i)) }
  }
  IF (size(newSet) $>$ MaxT) {
    newSet = SORT(newSet, SKLD(t_i), DESCENDING)
    newSet=newSet(1..MaxT)
  }
  RETURN newSet
}

3.4 Submitted runs

The parameters we used for the runs include:

- BM25 parameters: k1 = 3.44, b = 0.297
- Term selection function: Kullback-Leibler Distance (KLD)
- Term section threshold: 2.5
- Maximum number of terms included in each query: 7
- Weighting after selection: terms in the topic title are given extra weight by considering that they appear a number r_{Load} of times in a hypothetical relevance set of size R_{Load}. These numbers are free parameters, and are added to the observed r_i and R counts. In our runs we set these parameters to $R_{Load} = 50$ and $r_{load} = 49$.

The methods used in the various submitted runs are shown in Table 1.

Table 1: HARD Track: submitted runs

Run	Stem	Sn*	GP	BP	Desc
MSRCBaseline	yes	no	no	no	no
MSRCh4SD	yes	no	no	no	yes
MSRCh4SG	yes	no	yes	no	no
MSRCh4SGB	yes	no	yes	yes	no
MSRCh4SSn	yes	yes	no	no	no
MSRCh4SSnB	yes	yes	no	yes	no
MSRCh4SSnG	yes	yes	yes	no	no
MSRCh4SSnGB	yes	yes	yes	yes	no
MSRCh4SSnGBD	yes	yes	yes	yes	yes**

* Sn = Snippets; GP = Good Phrases; BP = Bad Phrases; Desc = Description
** We only used the description for queries with no user feedback (i.e. no snippets, good phrases or bad phrases).

3.5 Results

Results are shown in Tables 2 and 3.

The results may be summarised as follows. Query expansion from snippets selected by the user helped this year (in contrast to last year when we failed to get any benefit). Good phrases also helped, even in the relatively simple-minded way that we used them (just to add extra single terms). Bad phrases (in the way that we used them here) did not help; there is clearly much scope for looking at different ways of using this information.

4 WEB Track

In the Web Track we focus on three types of evidence:

Table 2: HARD Track, hard relevance evaluation

	AveP	P@10	RPrec
MSRCBaseline	0.2098	0.2733	0.2336
MSRCh4SD	0.2511	0.3533	0.2539
MSRCh4SG	0.2581	0.3400	0.2752
MSRCh4SGB	0.2585	0.3400	0.2763
MSRCh4SSn	0.2428	0.2622	0.2621
MSRCh4SSnB	0.2396	0.2489	0.2557
MSRCh4SSnG	0.2836	0.3044	0.2981
MSRCh4SSnGB	0.2839	0.3044	0.2992
MSRCh4SSnGBD	0.2841	0.3111	0.2966

Table 3: HARD Track, hard relevance evaluation

	AveP	P@10	RPrec
MSRCBaseline	0.2077	0.3444	0.2409
MSRCh4SD	0.2544	0.4133	0.2857
MSRCh4SG	0.2490	0.4333	0.2875
MSRCh4SGB	0.2506	0.4289	0.2889
MSRCh4SSn	0.2329	0.3311	0.2591
MSRCh4SSnB	0.2284	0.3178	0.2525
MSRCh4SSnG	0.2612	0.3911	0.2938
MSRCh4SSnGB	0.2615	0.3911	0.2938
MSRCh4SSnGBD	0.2631	0.3978	0.2955

1. Text: title, body and anchor.
2. Link recommendation: PageRank or ClickDistance (defined as the minimum number of hyper-links one needs to follow to go from `http://firstgov.gov` to the page).
3. URL depth: length of URL in characters.

For the text, we use a new variant of BM25 where weights and length normalisation parameters are distinct per-field (this is discussed in Section 4.1). For link-based and URL features we employ new combination functions, which we find by analysing the relevant and retrieved documents for a set of training queries (this is discussed in Section 4.2).

We deal with the mixed HP-NP-TD query stream through tuning. We have examples of all three types from TREC-2003, and so we conducted several tuning runs, looking at performances specifically to each task and overall (this is discussed in Section 4.3) The main difference we found across query types is that stemming helps TD and hurts the other two, so we vary the stemming across our runs. Our submission runs and results are discussed in Section 4.4.

4.1 Integrating Content Features Across Multiple Fields

We refer to the different annotated parts of a document, such as title and body, as *document fields*. Furthermore we use the term *anchor field* to refer to all the anchor text in the collection pointing to a particular document.

In previous work we showed that combining BM25 scores across fields can lead to a dangerous over-estimation of the importance of the term [7]. We proposed to combine the term-frequencies (weighting them accordingly to their field importance) and using the resulting *pseudo-frequency* in the BM25 ranking function. Furthermore we showed how to adapt automatically the parameters K_1 and B to changes in the weights.

In this submission we have modified slightly this approach to take into account fields of extremely different field lengths (such as those of the title and anchor). In particular, we have modified the function so that it can use a different length normalising factor B for every field-type.

This is done by computing a field-dependant normalised term-frequency:[1]

$$\bar{x}_{d,f,t} := \frac{x_{d,f,t}}{(1 + B_f(\frac{l_{d,f}}{l_f} - 1))}$$

$f \in \{$BODY, TITLE, ANCHOR$\}$ indicates the field type, $x_{d,f,t}$ is the term frequency of term t in the field type f of document d, $l_{d,f}$ is the length of that field, and l_f is the average field length for that field type. B_f is a field-dependant parameter similar to the B parameter in BM25. In particular, if $B_f = 0$ there is no normalisation and if $B_f = 1$ the frequency is completely normalised w.r.t. the average field length.

These term frequencies can then be combined in a linearly weighted sum to obtain the final term *pseudo-frequency*, which is then used in the usual BM25 saturating function. This leads the following ranking function, which we refer to as BM25F:

$$\bar{x}_{d,t} = \sum_f W_f \cdot \bar{x}_{d,f,t}$$

$$BM25F(d) := \sum_{t \in q \cap d} \frac{\bar{x}_{d,t}}{K_1 + \bar{x}_{d,t}} w_t^{(1)}$$

where $w_t^{(1)}$ is the usual RSJ relevance weight for term t, which reduces to an idf weight in the absence of relevance information (note that this does not use field information).

Note that this function requires one B_f and one W_f parameter per field, plus a single saturating parameter K_1. This constitutes a total of $(F * 2 + 1)$ parameters. Because of the dependence structure fo the parameters, we can brake down the global optimisation into smaller optimisation problems of one or two parameters. The optimisation procedure that we followed to is as follows:

- B_f: Independently for every field (setting the other field weights to zero), optimise B_f and $K1$.

- K_1: Setting all field weights W to 1, and using all the B_f values previously found, optimise $K1$.

- W_f: Setting the body weight to 1, the previously found values of B_f and $K1$, optimise the weights W_f for the title and anchor fields (adapting K_1 to each weight setting, as indicated in [7]).

[1] The equation for normalised term-frequency and that for BM25 contain errors in the printed version, which have been corrected in this online version.

Parameter	TD'03	NP'03
K_1	27.5	4.9
B_{TITLE}	0.95	0.6
B_{BODY}	0.7	0.5
B_{ANCHOR}	0.6	0.6
W_{TITLE}	38.4	13.5
W_{BODY}	1.0	1.0
W_{ANCHOR}	35	11.5

Table 4: BM25F parameters obtained by optimising the Topic Distillation (TD) and Name Page (NP) TREC tasks of 2003. See text for details.

This constitutes $(F + 1)$ optimisations in 2 dimensions (2D) and one optimisation in 1D. 2D and 1D optimisations were done by a robust line-search type algorithm, optimising Precision@10 on a set of training topics.

In order to evaluate our approach against standard BM25 and [7] we used 4-fold cross–validation over a set of TREC topics. We did this with the 2003 Name Page (NP) and 2003 Topic Distillation (TD) topics sets separately (each has 50 topics). A full optimisation run took approximately 3 days to complete (running on top of Keenbow without making any effort to optimise the code for this task). The resulting performances were significantly better for a range of measures, so we decided to use BM25F for our final submissions.

Since the variance of the parameters was not large across cross–validation sets, we used for our final submissions the average parameter values obtained in the cross-validation. The values of these parameters are indicated in table 4.

4.2 Integrating Document Features

Our new combination functions are suggested based on analysis of the relevant and retrieved documents for a set of training queries. Examining such sets is reminiscent of Singhal et al [8].

In general, we address the problem of choosing a score contribution function f for adding static scores S to existing document scores D, with $\mathit{CombinedScore} = D + f(S)$. As an illustration, we consider the case were D is BM25 and S is PageRank (see Figure 3).

For a set of training queries we identify two sets of documents. The relevant set R is the set of known relevant documents for the training queries. The retrieved set T is the set of top-ranked documents if we retrieve using D only. In this example we use the top r documents for each query, where

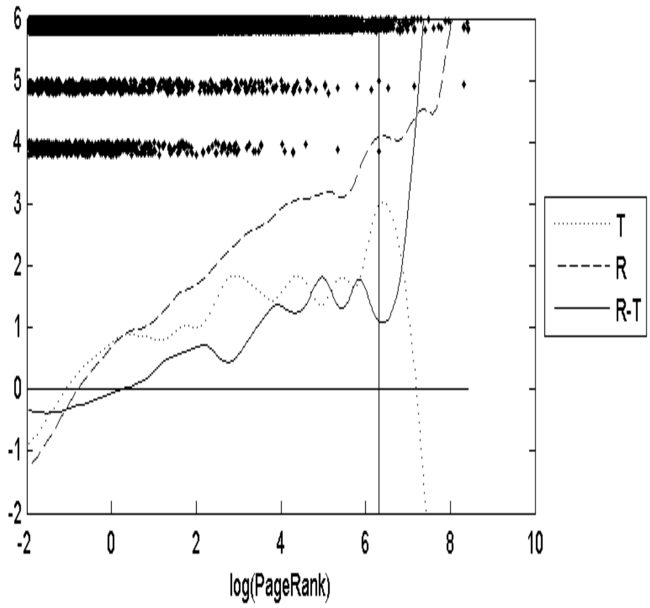

Figure 3: Web Track: Analysis for finding the score contribution function for PageRank. The horizontal axis is log(PageRank). The plots are $\log \frac{P(S=s|R)}{P(S=s|\bar{R})}$ (labelled as R), $\log \frac{P(S=s|T)}{P(S=s|\bar{T})}$ (labelled as T) and the difference between the two (R-T). The choice of score contribution function f is based on the shape of $R - T$.

r is the number of known-relevant documents for that query, which makes T the same size as R.

Assuming that D and S are independent, the correct score contribution for $S = s$ is $\log \frac{P(S=s|R)}{P(S=s|\bar{R})}$. In practice we approximate the set of irrelevant documents $\bar{R}$ using the whole collection C. Based on our training set, the score contribution is given by the line labelled R in Figure 3. The upwards slope of the line indicates that PageRank is a useful relevance indicator.

However, D and S are not independent, and we can see this if we plot T in the same manner (also Figure 3). The upwards slope of the line indicates that BM25 is already retrieving high-PageRank pages. The difference between the curves $R - T$ suggests the shape of f. We tried log PageRank, since $R - T$ looks quite straight in the region of interest:

$$f(PageRank) = w \log(PageRank) \quad (1)$$

We also tried a sigmoid of log PageRank, since $R - T$ does appear to flatten out. Tuning the sigmoid of log PageRank gave us our final $f(S)$ for PageRank:

$$f(PageRank) = \frac{w}{1 + e^{a(-\log(PageRank)+b)}} \quad (2)$$

although due to time constraints we fixed $a = 1$.

Similar analysis for $S = ClickDistance$ suggested:

$$f(ClickDist) = w \frac{k}{k + ClickDist} \quad (3)$$

Run	Average	TD MAP	NP MRR	HP MRR
MSRC04B1S	0.5392	0.159	0.719	0.741
MSRC04B2S	0.5461	0.162	0.731	0.745
MSRC04B1S2	0.4985	0.136	0.709	0.651
MSRC04B3S	0.4601	0.121	0.674	0.585
MSRC04C12	0.5458	0.165	0.724	0.749

Table 5: Web Track results. The 'average' is just an average of the other three columns.

Analysis for $D = BM25 + PageRank$ and $S = URLLength$ suggested a similar function:

$$f(URLLength) = w \frac{k}{k + URLLength} \quad (4)$$

and this performed better than the negative linear function used in previous experiments.

4.3 Preliminary Experiments

To evaluate the combination of content and link features we used a mixed set of 120 TREC 2003 Web queries (40 from TD, 40 from NP and 40 from Home Page (HP)). We used the BM25F parameters obtained optimising Prec@10 on TD and NP. We tuned the PageRank (PR) weights w and b (keeping $a = 1$) and similarly we tuned k and w for ClickDistance (CD). URL-length (URLl) parameters were tuned for PageRank and ClickDistance afterwards on the same set (we checked the risk of over-fitting by validating on the remaining 2003 queries).

Finally, we tested several rank combination methods without much success. The only interesting result we found was to interleave the documents of two disparate runs (removing duplicates from the bottom).

4.4 Submitted Runs and Results

Our submission runs were:

- MSRC04B1S: BM25 NP tuning + PR + URLl
- MSRC04B2S: BM25 NP tuning stem + PR + URLl
- MSRC04B1S2: BM25 NP tuning + CD + URLl
- MSRC04B3S: BM25 TD tuning + CD
- MSRC04C12: interleave MSRC04B1S and MSRC04B2S

Our results are summarised in Table 5. Based on our training runs we expected stemming to help TD and hurt the other two. Actually the first two lines of the table show that turning on stemming helped slightly across all tasks. Since the differences were small, interleaving the two (MSRC04C12) only had the effect of adding some (slightly positive) noise. The runs with click distance were somewhat worse than the PageRank runs, but they performed two do similar jobs.

5 Conclusions

In the HARD Track, we have succeeded in revising our relevance feedback mechanisms (in particular, the rules for query expansion) in such a way that we can benefit from user feedback on snippets. Other matters have been left in abeyance, including (a) the use of active learning techniques for choosing the snippets to present to the user (as we did last year), (b) the use of negative information (words or phrases specified by the user as negative indicators of relevance), and (c) passage retrieval.

In the Web Track we did not find new forms of evidence or new ways of dealing with mixed query streams. Instead we concentrated on dealing well with the different text fields in BM25 and adding static information using appropriate functions. It appears that there was still room for gains in these areas, and our final results were very satisfactory.

References

[1] C Carpineto, R De Moro, G Romano, and B Bigi. An information-theoretic perspective to automatic query expansion. *ACM Transactions on Information Systems*, 19:1–27, 2001.

[2] S E Robertson. On term selection for query expansion. *Journal of Documentation*, 46:359–364, 1990.

[3] 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)*, NIST Special Publication 500-246, pages 151–162. Gaithersburg, MD: NIST, 2000.

[4] S E Robertson and S Walker. Threshold setting in adaptive filtering. *Journal of Documentation*, 56:312–331, 2000.

[5] 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*, NIST Special Publication 500-251, pages 439–446. Gaithersburg, MD: NIST, 2003.

[6] S E Robertson, H Zaragoza, and M Taylor. Microsoft Cambridge at TREC 2003: Hard track. In E M Voorhees and L P Buckland, editors, *The Twelfth Text REtrieval Conference, TREC 2003*, NIST Special Publication 500-255, pages 418–425. Gaithersburg, MD: NIST, 2004.

[7] S E Robertson, H Zaragoza, and M Taylor. Simple BM25 Extension to Multiple Weighted Fields. Thirteenth Conference on Information and Knowledge Management, CIKM 2004.

[8] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM Press, 1996.