Microsoft Cambridge at TREC–14: Enterprise track

Nick Craswell
Microsoft Research Ltd
7 J.J.Thomson Avenue
Cambridge CB3 0FB, UK
nickcr@microsoft.com

Hugo Zaragoza
[Microsoft Research Ltd]
Now at Yahoo! Research
Barcelona
hugoz@es.yahoo-inc.com

Stephen Robertson
Microsoft Research Ltd
7 J.J.Thomson Avenue
Cambridge CB3 0FB, UK
ser@microsoft.com

1 Summary

A major focus of much work of the group (as it has been since the City University Okapi work) is the development and refinement of basic ranking algorithms. The workhorse remains the BM25 algorithm; recently [3, 4] we introduced a field-weighted version of this, allowing differential treatment of different fields in the original documents, such as title, anchor text, body text. We have also recently [2] been working on ways of analysing the possible contributions of static (query-independent) evidence, and of incorporating them into the scoring/ranking algorithm. Finally, we have been working on ways of tuning the resulting ranking functions, since each elaboration tends to introduce one or more new free parameters which have to be set through tuning.

We used all these techniques successfully in our contribution to the Web track in TREC 2004 [4]. This year's relatively modest TREC effort is confined to applying essentially the same techniques to rather different data, in the Enterprise Track's known item (KI) and discussion search (DS) experiments. The main interest is whether we can identify some fields and features that lead to an improvement over a flat-text baseline, and as a side effect to verify that our ranking model can deliver the benefit.

2 Ranking model and implementation

This section describes: 1) BM25 with field weighting and per-field length normalisation (BM25F), 2) Incorporation of query-independent, non-text features via linear combination and 3) The tuning and ranking framework we used this year.

To calculate BM25F [4] we first calculate a normalised term frequency for each field:

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

$f \in \{$SUBJECT, BODY, QUOTED$\}$ 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*:

$$\bar{x}_{d,t} = \sum_f W_f \cdot \bar{x}_{d,f,t} \tag{2}$$

with weight parameters W_f. This is then used in the usual BM25 saturating function. This leads the following ranking function, which we refer to as BM25F:

$$BM25F(d) := \sum_{t \in q \cap d} \frac{\bar{x}_{d,t}}{K_1 + \bar{x}_{d,t}} \, w_t^{(1)} \tag{3}$$

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).

Static features are combined via linear combination with the BM25 score as in [2]. They are either added with a single weight, or with a three-parameter transformation. This transformation is essentially a weighted sigmoid, but its formulation depends on the nature of the feature variable. For a feature which is constrained to be non-negative and has a natural zero, such as a count, the transformation is:

$$w \frac{x^a}{k^a + x^a} \tag{4}$$

(note that this is a generalised version of the tf function in BM25). For a feature which does not have a natural zero (for example, Year), the transformation is:

$$w \frac{e^{a(x+b)}}{1 + e^{a(x+b)}} \tag{5}$$

These are equivalent under a log transformation of the original feature. In order to reduce the parameter space, we used a single weight where possible, for example for binary features like #parents. We used the three-parameter combination when the single-weight combination was clearly sub-optimal. In the event, the only result using a three-parameter transformation reported below applies the second form (5) to Year.

Our ranking system was implemented based on tables of statistics rather than an inverted index, extracted and combined via one-off scripts. This gave us maximum flexibility for extraction of fields and statistics. We tuned on the Known Item experiment's 25 query training set, using two separate approaches: iterative 1-D explorations of the parameter space [4] and gradient descent [1]. Exploring the parameter space in two ways gave us greater confidence in tunings. Although extensive tuning without a test set can lead to overfitting, we believe that our system was sufficiently constrained to avoid overfitting, so we were happy to stick with the tuning that maximised MRR.

3 Fields and features

We eliminated documents, such as index pages, that were not messages. From each message we extracted three text fields and multiple query-independent statistics. The fields were:

Subject The text on the subject line of the message. Except in MSRCKI5 and MSRCDS5, where we combined Subject and From lines into this field.

Body The unquoted, new content of this message.

Quoted The quoted content from a previous message.

We would expect the subject text to be most important in describing a message, and perhaps the quoted text to be the least important.

We extracted several static features. We describe all of them, but only the first three were used in submitted runs:

#parents Has value 1 if the message is in reply to another message, 0 otherwise. KI messages tend to be at the start of a thread, so have #parents=0.

Year The year in which the message was sent e.g. 2001. In KI there was some benefit in preferring new messages.

RE The term frequency of the string 'RE' in the subject line. Since #parents is a strong but noisy feature, we decided to add this feature which is similar but noisy in a different way. For example, a message might have #parents=0 due to a problem with resolving the thread tree, but have RE=1. Or a message with #parents=1 might actually be the start of a new discussion, whose subject line was edited and therefore has RE=0.

URLs The number of URLs in the message body. We noticed that important messages might be announcing a web site URL.

#children, #ancestors, #descendants These tell us about the thread tree, similarly to #parents, except were less useful in ranking. For example, if a message is 5th in line in a linear thread then it would have #ancestors=4. If it has three replies then #children=3.

Date The UNIX date, in seconds since the epoch. Like Year but with finer granularity.

Table 1: The main tuning used in MSRCKI1 and MSRCDS1.

K_1	2.0				
$W_{Subject}$	20.0	$B_{Subject}$	0.6		
W_{Body}	0.68	B_{Body}	0.03		
W_{Quoted}	0.7	B_{Quoted}	0.8		
$w_{\#parents}$	-2.5				
w_{Year}	6.0	a_{Year}	0.3	b_{Year}	-2005

Prolificauthor The count of messages from the same author. We thought that there might be some relationship between how often an author posts and the importance of their messages.

We tried static features one at a time, to get an idea of which ones worked best. Then we added our best 2 static features (Year and #parents), and retuned all parameters. This gave a worthwhile improvement in MRR. Adding the best 4 static features (Year, #parents, URLs and RE) the result was only marginally better. We did not try adding more than 4 at once.

4 Tuning and results

We have 5 different tunings that were used for our submissions in both tasks. The first tuning uses three text fields and the two best static features. To see if there is any benefit from these, we remove the static features in the second tuning and use a uniform tuning of BM25F in the third. In the fourth and fifth tuning we add some elements were more 'risky': two additional static features (URLs, RE) and then the augmented Subject field that also contains the From line. In the event, the tuning gave 'URLs' a weight of zero, so the resulting ranking function uses only three static features.

The most tuning effort, via exploration and gradient descent, was applied to the first tuning (Table 1). The second and third tunings involved zeroing the static feature weights and making the BM25F tuning uniform ($W_* = 1$, $B_* = 0.8$). The fourth tuning involved an extra feature weight ($w_{RE} = -1.12$) and a slightly modified BM25F tuning, but the weights of #parents and Year remained unchanged. The fifth tuning was identical to the fourth, with the difference being the extra text in the Subject field. In summary, the carefully explored tuning space was the first tuning, and the others are minimal modifications.

Table 2 shows that there were good gains to be had, in both training and test, from the field weighting and static features. The more speculative tunings (4 and 5) showed further improvement on the test set, even though they were not strong in training. In fact our best result on test (second best of all submissions to the task) was tuning 5, which had seemed worse than tunings 1 and 4 on training. In the absence of training data, we submitted to the DS task with the same tunings (Table 3). We suspected that our KI priors for finding messages at the head of a thread would be harmful in the DS task and this

Table 2: KI training and test results.

	MRR			
Run	Train	Test	Fields	Static
MSRCKI1	0.695	0.546	Subject,Body,Quoted	Year, #parents
MSRCKI2	0.64	0.522	Subject,Body,Quoted	None
MSRCKI3	0.457	0.458	Uniformly weighted	None
MSRCKI4	0.699	0.563	Subject,Body,Quoted	Year, #parents, RE
MSRCKI5	0.673	0.613	Subject+From,Body,Quoted	Year, #parents, RE

Table 3: DS test results.

Run	MAP	Fields	Static
MSRCDS1	0.2802	Subject,Body,Quoted	Year, #parents
MSRCDS2	0.3139	Subject,Body,Quoted	None
MSRCDS3	0.3042	Uniformly weighted	None
MSRCDS4	0.2696	Subject,Body,Quoted	Year, #parents, RE
MSRCDS5	0.2713	Subject+From,Body,Quoted	Year, #parents, RE

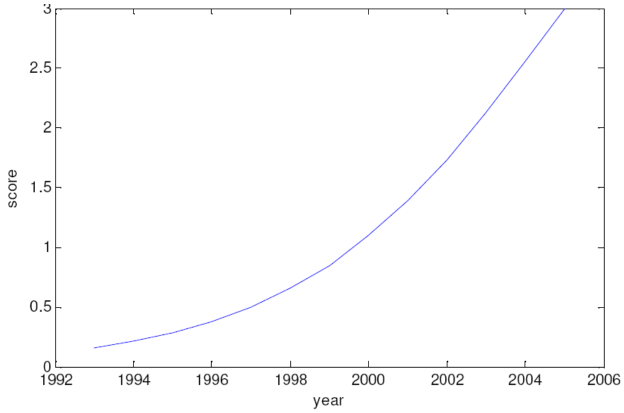

Figure 1: Scoring function for the query-independent feature Year.

proved to be the case, with the best performance coming from MSRCDS2: field weighting but no static features.

As indicated above, in our submitted runs Year is the only feature to which we applied a sigmoid transformation (equation 5). This makes for an interesting observation: the value of b resulting from tuning is -2005 (see Table 1). In effect, the system has learnt that the feature might be better defined as ‘age’ rather than Year! The effective shape given by the sigmoid is in Figure 1. This may also explain why we got less benefit from the Date feature. If Year needs $b = -2005$ then Date needs roughly $b = -1104537600$[1], and our tuner may not have explored that region of the parameter space.

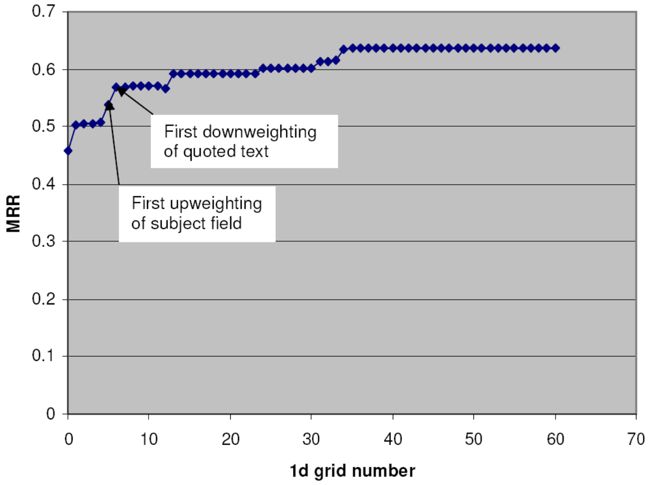

Figure 2: Ten rounds of tuning the six BM25F parameters gave these 60 MRR values on the training set (and the leftmost untuned point). The tuning order was: B_{Body}, $B_{Subject}$, B_{Quoted}, W_{Body}, $W_{Subject}$, W_{Quoted}

[1] Sat, 1 Jan 2005 00:00:00 UTC is 1104537600 seconds since the standard epoch of 1/1/1970.

5 Conclusion

We applied our Web Track 2004 ranking model to email search. We identified three text fields: subject, body and quoted. By treating each of these differently we saw some gain in the KI experiment, over a uniformly weighted baseline. We also identified a number of query-independent statistics. Factoring these into the ranking gave us further gains on KI, although we did not add all our features at once. The field weighting performance gains in KI were also there in DS, but the static features harmed our DS effectiveness. This suggests that a good KI result, perhaps a recent message at the head of a thread, is different from a good DS result. Given the gains in effectiveness over a flat text baseline, we can claim some success in transferring our Web retrieval techniques into a different application domain. However, the importance of having appropriate training data (lacking for DS) is emphasised.

References

[1] C J C Burges, T Shaked, E Renshaw, et al. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, 2005.

[2] Nick Craswell, Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Relevance weighting for query independent evidence. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 416–423, New York, NY, USA, 2005. ACM Press.

[3] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 extension to multiple weighted fields. In *CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management*, pages 42–49, New York, NY, USA, 2004. ACM Press.

[4] Hugo Zaragoza, Nick Craswell, Michael Taylor, Suchi Saria, and Stephen Robertson. Microsoft Cambridge at TREC-13: Web and HARD tracks. In *Proceedings of TREC-2004*, 2004.