Relevance Weighting for Query Independent Evidence

Nick Craswell, Stephen Robertson, Hugo Zaragoza and Michael Taylor
Microsoft Research
Cambridge, U.K.
{nickcr,ser,hugoz,mitaylor}@microsoft.com

ABSTRACT

A query independent feature, relating perhaps to document content, linkage or usage, can be transformed into a static, per-document relevance weight for use in ranking. The challenge is to find a good function to transform feature values into relevance scores. This paper presents FLOE, a simple density analysis method for modelling the shape of the transformation required, based on training data and without assuming independence between feature and baseline. For a new query independent feature, it addresses the questions: is it required for ranking, what sort of transformation is appropriate and, after adding it, how successful was the chosen transformation? Based on this we apply sigmoid transformations to PageRank, indegree, URL Length and ClickDistance, tested in combination with a BM25 baseline.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]

General Terms

Experimentation

Keywords

Web Search, Ranking, Probabilistic IR

1. INTRODUCTION

A relevance weighting scheme for query independent evidence is one way of combining a query independent feature such as Google's PageRank [3], with a query dependent baseline. The idea is to attach a static relevance weight to each document, based on the feature. This weight is then linearly combined with the query dependent baseline score, to give a new score and ranking. The challenge is to choose a good method for transforming the feature into a per-document relevance weight to enhance retrieval effectiveness. In particular we focus on combination with a BM25 baseline.

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

We are motivated by the wealth of potentially useful static ranking features. It is easy to think of a number of examples. PageRank may indicate whether a page is a good Web search result. When searching for technical answers in a newsgroup, feature(s) indicating a message's position in the thread might be useful (since the root usually contains the question, and later messages the answer). File creation or modification date may be an important search feature when searching a Personal Computer, if the user is more often searching for recent files. A lower-priced product, but not too low-priced, might be a better answer in e-commerce search. In all these cases, the question is whether the feature can be used to adjust the ranking and improve search effectiveness, and if so how best to incorporate it.

Without some analysis, the answer is not immediately obvious. It is possible to guess a combination function and see if it works. If the guessed combination fails, this does not mean the feature is useless. If the guessed combination succeeds, this does not mean it is optimal. Further, the appropriate adjustment depends on the baseline ranking. A query independent feature which indicates relevance might not be needed at all, if the baseline already makes use of highly correlated features.

Our approach is to model the appropriate static score, for a given static feature, corpus and baseline ranking. Modelling this weight can tell us a number of things:

1. Whether the feature is needed. The model can predict that no adjustment is needed, making it a light-weight prediction of whether a feature is worth adding to an existing system.

2. What shape the adjustment should take. This helps us choose an appropriate functional form, for transforming the feature into a relevance weight.

3. Whether mistakes have been made in combination. After combination, the model should predict that no further adjustment is necessary, making it a light-weight test for combination errors.

Our model correctly predicts that PageRank, link indegree, URL length and ClickDistance (described in Section 4) are each useful static features when searching the TREC .GOV test collection from a BM25 baseline. It also suggests that sigmoid transformations are appropriate for turning the features into relevance weights. After applying a single feature, it predicts no further adjustment for that feature is necessary (indicating the combination was successful) except in the case of ClickDistance. After adding PageRank, it cor-

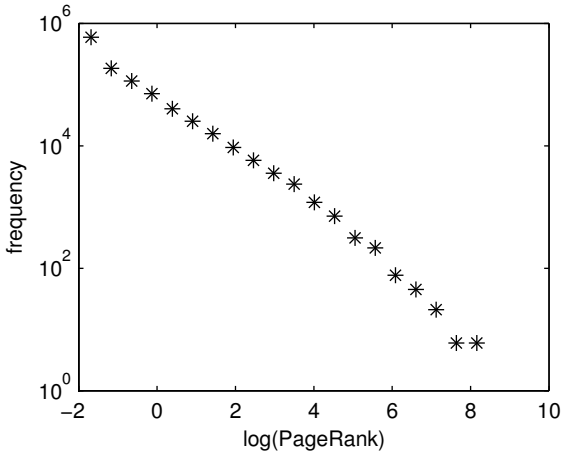

Figure 1: For the .GOV test collection, PageRank has a power law distribution.

rectly predicts that URL length is still useful, and that indegree and ClickDistance are no longer needed.

First we survey past work on query independent evidence and its use in ranking. Then we evaluate a number of relevance weightings for PageRank and use this initial experiment to motivate the new approach. After describing the new approach we apply it to link indegree, URL Length and ClickDistance.

2. QUERY INDEPENDENT EVIDENCE AND COMBINATION

In a ranking scheme, some features pertain to query terms, so their usage depends on the current query. For example, in most ranking schemes we do not know which term frequency statistics will be used until we know the query. Other features, we know we will use before we know the query, and in that case we refer to them as query independent evidence.

A well known example of query independent evidence is PageRank [3], a score assigned to each document in a Web crawl. It indicates how easy it would be to reach that page by randomly following links: the "random surfer" model. A page can have higher PageRank if more pages link to it, or if the linking pages themselves have higher PageRank or lower out-degree, because each of these increases the chances of the random surfer reaching the page. In this paper we use standard PageRank, with a random jump probability of 1/7 and a mean PageRank value of 1.

The initial Google paper [3] did not describe how PageRank should be combined with a query dependent baseline. Combination is difficult because PageRank has a power law distribution [10]. This means a simple linear combination of scores, for example, would lead to most pages getting almost no score and a few getting a very large score. For example, calculating PageRank on TREC .GOV such that the average PageRank is 1 gave the distribution in Figure 1. The top PageRank is 4522.6, which is 21,084 times the median PageRank of 0.2145.

Simple link counts may also be used as query independent evidence, such as a page's indegree or outdegree. Further metrics can be found by collapsing all of a host's nodes into a single node, and calculating host indegree, host outdegree and host PageRank (HostRank) [1]. URLs are also a source of useful static features, preferring short URLs or dividing URLs into different types (root, subroot, path and file)[9].

Non-web environments may also have query independent evidence which is worth exploiting, for example thread size, number of replies and message date might be useful static ranking features in a message archive. Query independent document usage information — such as aggregate clickthrough, visit frequency or dwell time statistics — may also prove useful in ranking. The approach described in this paper may be applied to any of these new features.

Document length is used in many retrieval models; it is used in every query, therefore in some sense it is a form of query independent evidence. However, it is usually applied at the core of the query dependent retrieval model, for per-term normalization. This makes document length quite different from the other static features we consider, and we do not survey here different forms of length normalisation. Other papers studied length normalization [12, 9, 7].

There are three broad approaches to combining static features and a query dependent baseline ranking: rank-based, as a language modelling prior and as a relevance score adjustment.

Rank-based combination has been used with some success [6, 4, 14]. This involves turning the baseline and static scores into two rankings, and combining based solely on ranks. This has the advantage of ignoring the score distributions, so for example it is impossible to fall foul of PageRank's power law distribution by giving too much of a boost to pages with very high PageRank. However, it should be possible to do as well or better using scores, since they contain more information. For example it is possible to generate document ranks given scores but not vice versa.

In a language modelling framework, prior probabilities were calculated for page length, indegree and URL type in [9]. For example, indegree was divided into bins on a log scale, and for each bin a prior was calculated. This was then combined with the language modelling probability (a multiplicative combination).

The other approach, and the one used here, is to rank based on a linear combination of baseline and static scores. This can be done using raw scores, for example BM25 with URL prior and PageRank [8]. However, it may be possible to do even better with a nonlinear transformation of the input feature. For example, using log(indegree) rather than raw indegree [13].

3. STATIC RELEVANCE WEIGHTS AND DENSITY ANALYSIS

We introduce our approach with a detailed case study of PageRank relevance weighting in the TREC .GOV test collection. PageRank combination is a well known and difficult problem, so an obvious choice here. Our baseline ranking is BM25 with field weighting [11] and per-field length normalisation parameters [15]. We use three text fields: body, title and referring anchor text.

The case study has three parts. First we explore the notion of relevance weighting for PageRank, by choosing three PageRank weighting functions, tuning their parameters and evaluating their effectiveness. This gives us a preferred method for adding PageRank, but not a reason for doing it that way. In the second part we develop a model for

the appropriate adjustment, assuming independence. This is not a good match for the weighting found empirically, so we finally use a simple heuristic we call FLOE (feature's log odds estimate) for taking dependence into account.

3.1 Static relevance weighting

BM25, like a number of other ranking approaches, produces a relevance score which is the sum of weights for each query term. In such a system it is natural to consider adding a further weight which depends on some query independent document feature. The question is how to turn a static feature into an appropriate relevance weight. In our initial experiment, we try three different transformations on PageRank, tuning their parameters.

Our tuning set comprises 120 mixed topics from the TREC-2003 Web Track: 40 topic distillation, 40 homepage and 40 named page. Our test set is the full set of 225 mixed queries from the TREC-2004 Web Track [5]. In each case we rerank the top 1000 of our BM25 baseline using $finalscore = BM25score + PageRankWeight$. Limiting the reranking to the top 1000 is more efficient than reranking all documents with nonzero BM25 score, without making a large difference to system effectiveness.

Our first weighting function is:

$$log(S, w) = w \cdot \log(S) \tag{1}$$

where S is the value of the static feature (PageRank) and w is the weight we use when combining with BM25. Tuning involved an exploration of w from 0 in steps of 0.01, in order to maximise mean average precision (MAP) on the training set (Figure 2). The best tuning, at $w = 0.20$, gave MAP of 0.508 on the test set, from a baseline of 0.430. Note, we performed a similar experiment without taking log. The result was a very low weight ($w = 0.005$) and very poor test set performance of 0.439.

BM25's function for term frequency is of the form $\frac{TF}{k+TF}$. This function saturates, in the sense that it approaches 1 as TF increases (TF, like the other features discussed in this paper, is non-negative). Our second weighting function for PageRank, named *satu*, is similar:

$$satu(S, w, k) = w\frac{S}{k + S} \tag{2}$$

There are two parameters: w is the maximum which is approached as S increases and k which is the value of S where *satu* is $w/2$. Tuning via extensive 2d exploration gave $w = 1.34$ and $k = 1.36$, and test MAP of 0.515 (better than *log*).

The function *satu* can be reformulated as a sigmoid on $\log(S)$; however, it is not the most general sigmoid. By introducing another parameter we gain more control over the rate of saturation, making the function more general and flexible. This third function, known as *sigm*, is defined as follows:

$$sigm(S, w, k, a) = w\frac{S^a}{k^a + S^a} \tag{3}$$

Adding the extra parameter allows slightly better performance again: MAP of 0.523 with parameters $w = 1.8$, $k = 1$ and $a = 0.6$. For the sigmoid we only tune to one decimal place, since we are now working in three dimensions.

The three weightings (Figure 3) have remarkably similar slopes in the range -2..2, where 98% of pages in the collection lie. Note, there are gaps between the parallel line sections,

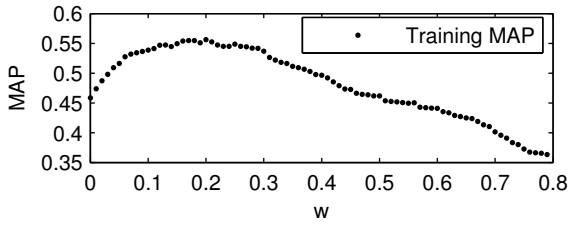

Figure 2: Tuning w, for *log* PageRank combination.

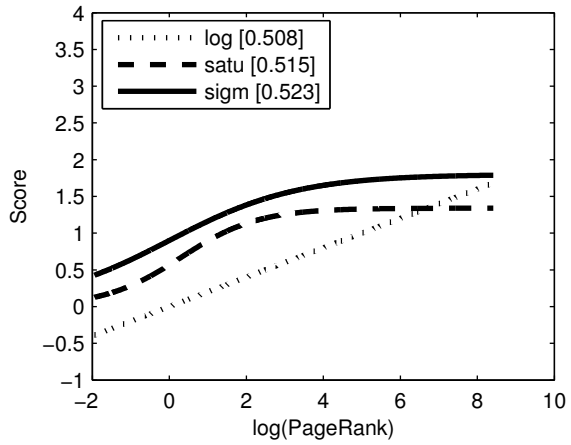

Figure 3: Weighting functions for PageRank tuned on a mixed set of WT03 queries. Numbers in square brackets are average precision on WT04, from a baseline run scoring 0.430. Functions have similar slopes in the range -2..2, where 98% of the corpus lies.

but these gaps could be closed by adding a constant to the weighting, which would have no effect on the overall ranking i.e. it is the slope not the height of a line which is important.

The similarities suggest there is some underlying truth, or 'ideal weighting' for PageRank in this experiment, and that this is particularly important in the range -2..2. However, there are a number of problems. We can not tell how close our guesses are to the ideal (or see an estimate of the ideal function), we can not analyse where these went wrong, if at all, and we do not know if there are other transformations that are even better. Our method of density analysis described in the remainder of this section attempts to address these issues.

3.2 Score adjustment assuming independence

We wish to rank according to the probability that a document (D) is relevant (R) for the current query: $P(R|D)$. When working with BM25 we rewrite this, in a way which preserves the rank order with respect to the query (Q):

$$P(R|D) \overset{Q}{\propto} \log \frac{P(R|D)}{P(\bar{R}|D)} \overset{Q}{\propto} \log \frac{P(D|R)}{P(D|\bar{R})} \overset{Q}{\propto} BM25(D) \tag{4}$$

If we then consider that a document D has two components, its content match M and its static score S, we can separate

it into two additive scores:

$$\log \frac{P(D|R)}{P(D|\bar{R})} = \log \frac{P(M,S|R)}{P(M,S|\bar{R})} \quad (5)$$

$$= \log \frac{P(M|R)}{P(M|\bar{R})} + \log \frac{P(S|M,R)}{P(S|M,\bar{R})} \quad (6)$$

$$\overset{Q}{\propto} BM25 + \log \frac{P(S|M,R)}{P(S|M,\bar{R})} \quad (7)$$

So BM25 can be linearly combined with a term that depends on modelling S with respect to M and R.

One possibility at this point is to drop the M term, assuming that content match and static score are independent. Under this independence assumption the correct BM25 score adjustment would be:

$$indep(S,R) = \log \frac{P(S|R)}{P(S|\bar{R})} \quad (8)$$

We can now plot *indep* and see if it matches our empirical score adjustments from Figure 3. Our plot uses kernel density estimation [2], which gives us smooth curves. However, we have done so equally successfully using histograms. We find the set of all relevant documents[1] for our 120 training queries, and denote such documents as R. We find the density, over log PageRank, of two sets of documents: the relevant documents $P(S|R)$ and the collection $P(S) \approx P(S|\bar{R})$. These are in Figure 4. Our adjustment *indep* is log of the ratio of these two lines. This can then be plotted in comparison to our best empirical adjustment.

Unfortunately, the result in Figure 5 is a curve which is much steeper than our empirical line. It has a slope of about 0.6, which would give quite bad performance compared to our training optimum of 0.2 (Figure 2). Note, there are very few relevant examples at the right-hand end of the plot (indicated by points). In this region the density estimate is less trustworthy.

3.3 FLOE: Feature's log odds estimate

We hypothesise that the mismatch in Figure 5 is due to the independence assumption. That is, BM25 is already retrieving high PageRank documents to some extent, so adjustment *indep* overstates the score boost for high-PageRank pages, in some sense double-counting PageRank. This is not surprising, since PageRank is a link metric and links are also used in the anchor text field of our baseline. A page with many incoming links is likely to have a higher PageRank and higher BM25 score if there is an anchor match.

FLOE (feature's log odds estimate) is one possible solution to avoid double-counting. It involves finding an estimate of the levels of S already in the baseline, using the *retrieved set*. We include in the set the top r baseline results for each query, where r is the number of known relevant results for that query. Similarly to R and $\bar{R}$, we denote documents in this set as T and all other documents $\bar{T}$. Then we can find a density estimate as we did for the relevant set: $\log \frac{P(S|T)}{P(S|\bar{T})} \approx \log \frac{P(S|T)}{P(S)}$.

The upward slope of this line in Figure 6 indicates that BM25 already has a strong tendency to retrieve high-PageRank pages. Now that we have a model for the PageRank levels

[1]This can be thought of as a set of document query pairs, since we allow a document to appear multiple times if it is relevant for multiple queries.

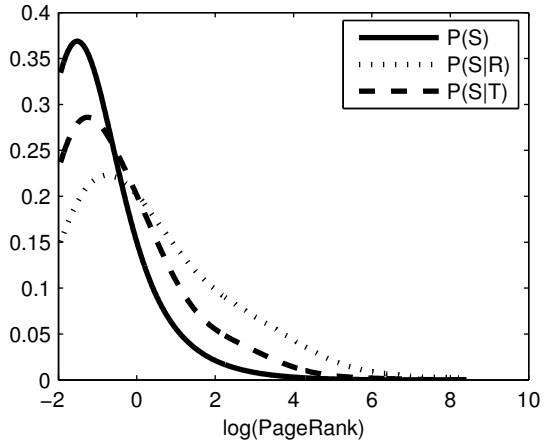

Figure 4: These are the density estimates used to model the curves in Figures 5 and 6.

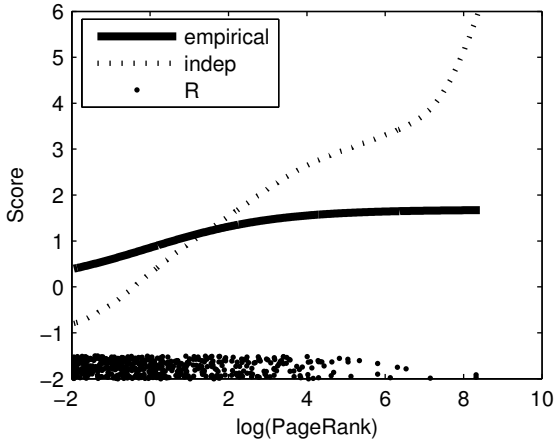

Figure 5: The estimated adjustment assuming independence (*indep*) does not match the empirical line (*sigm*) from Figure 3. The points labelled R are those in the relevant set we use for estimation.

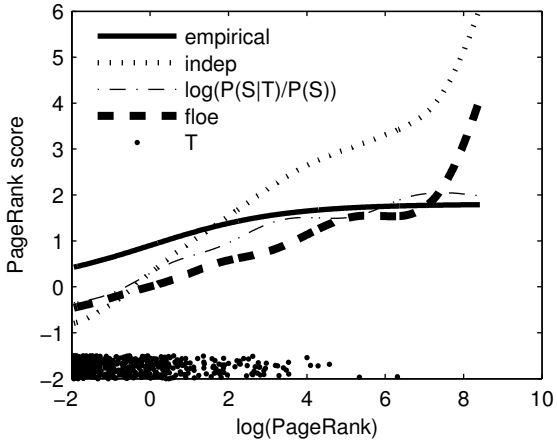

Figure 6: The *floe* line is a better match for the empirical line. The points labelled T are those in our retrieved set. Where data becomes sparse, our estimates break down.

in our baseline, we can remove this from *indep* by taking the difference of the two lines:

$$\mathit{floe}(S,R,T) = \log\frac{P(S|R)}{P(S|\bar{R})} - \log\frac{P(S|T)}{P(S|\bar{T})} \approx \log\frac{P(S|R)}{P(S|T)} \quad (9)$$

The simplification on the right hand side takes into account that $\bar{R} \approx \bar{T}$ since both are approximately the whole corpus.

In Figure 6 the line given by *floe* has a slope closely matching our empirical line in the region -2..2. It is giving a much more accurate estimate of our best PageRank adjustment than *indep*.

Note, our estimate becomes unreliable where data is sparse, so in all subsequent plots of equation (9) we limit the range to the range of documents in the retrieved set. The estimate is already unreliable at the right-hand end of this range, where there are only a few points, but beyond this range it becomes meaningless. This is a limitation of the model in the presence of sparse data, which could perhaps be ameliorated if the training set were larger. In a number of plots we will see bumps and other shapes where data become sparse.

A well known previous study [12] also compared the relevant and retrieved sets, and indeed prompted us to identify a retrieved set, in order to correct *indep*. Singhal, Buckley and Mitra analysed document length normalisation intensively, whereas this paper suggests that this type of approach can be used with PageRank and other static features. Also, we attempt to demonstrate that the difference between our lines (equation 9) can be used as a model for the appropriate static weight. By contrast, using the bucketing and plotting methods of [12] we would give downward sloping lines. It would indicate a gap, but give less information about the appropriate adjustment.

4. FINDING FUNCTIONAL FORMS

In our first experiments we guessed functions for transforming PageRank into a relevance weight. Now we have a method for estimating the appropriate adjustment, we can apply it for new features: link indegree, URL length in characters and ClickDistance. ClickDistance is a link metric which measures the page's minimum link distance from some root. In the case of .GOV we used firstgov.gov as the root, so pages it links to are at distance 1 and unseen pages linked to by them are distance 2 and so on. We gave documents which were unreachable the median ClickDistance of the reachable pages (=7).

Using FLOE we can plot score adjustment estimates for each feature (Figure 7). We have one parameter, the density estimate's kernel width (h), so we show three estimates in each plot. For smaller h, we see bumps and shapes which may be noise in our sample, rather than indicative for choosing a functional form. If we increase the width, we see smoother shapes emerging. All other plots in the paper use a kernel width (h) equal to 10% of the range of the retrieved set.

The estimated score adjustment is increasing for indegree, and decreasing for URL length and ClickDistance. In each case we see a flattening at the right-hand end of the plot, although this is also where in each case the data becomes sparse so we would expect our estimates to be less accurate. Nevertheless, the shapes suggest further use of sigmoid functions (strictly, sigmoids of $\log(S)$). We use a function with

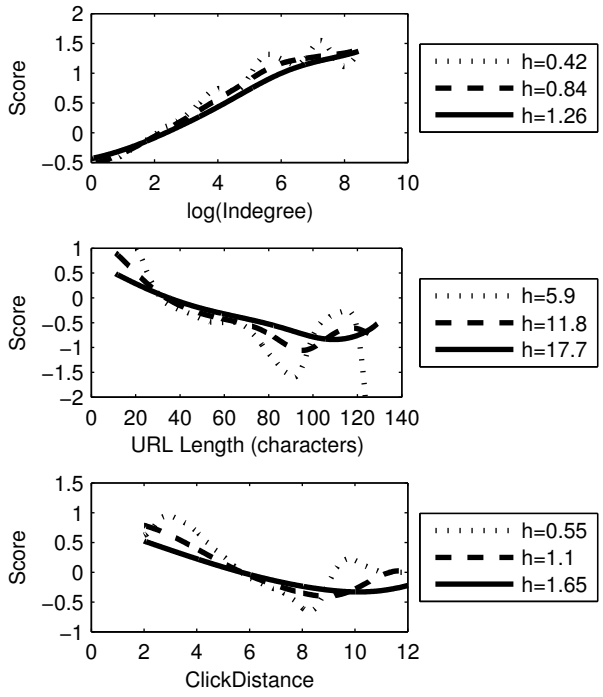

Figure 7: Equation (9) adjustment for indegree, URL length and ClickDistance on the mixed set of 120 training queries. In each case we plot for three different kernel widths (h), of 5%, 10% and 15% of the data range of the retrieved set.

downward slope:

$$w\frac{k^a}{k^a + S^a} \quad (10)$$

when S is URL length or ClickDistance, and one with upward slope:

$$w\frac{S^a}{k^a + S^a} \quad (11)$$

when S is Indegree (as used for PageRank). Each has three tunable parameters w, k and a.

Using these functions and tuning on our training set gives us the empirical lines in Figure 8. In each case in Figure 8, the *floe* line gives us a much better explanation of the appropriate relevance weighting than the *indep* line. The fact that this happens repeatedly gives us more confidence that estimate (9) is informative.

Effectiveness results for these tuned sigmoids are in Table 1. This repeats the result from Figure 3 that the sigmoid function had best performance for PageRank. It also indicates which static features worked best on our test set: PageRank > Indegree > URL Length > ClickDistance. These results are in keeping with recent TREC results [5]: URL length is less useful in tests involving named page queries, and the query-independent link features used by the top 4 groups were PageRank, HostRank, nothing and indegree.

5. POST-COMBINATION CHECKING

Another way of using FLOE is to assess the quality of combination. We do post-combination plots on the training

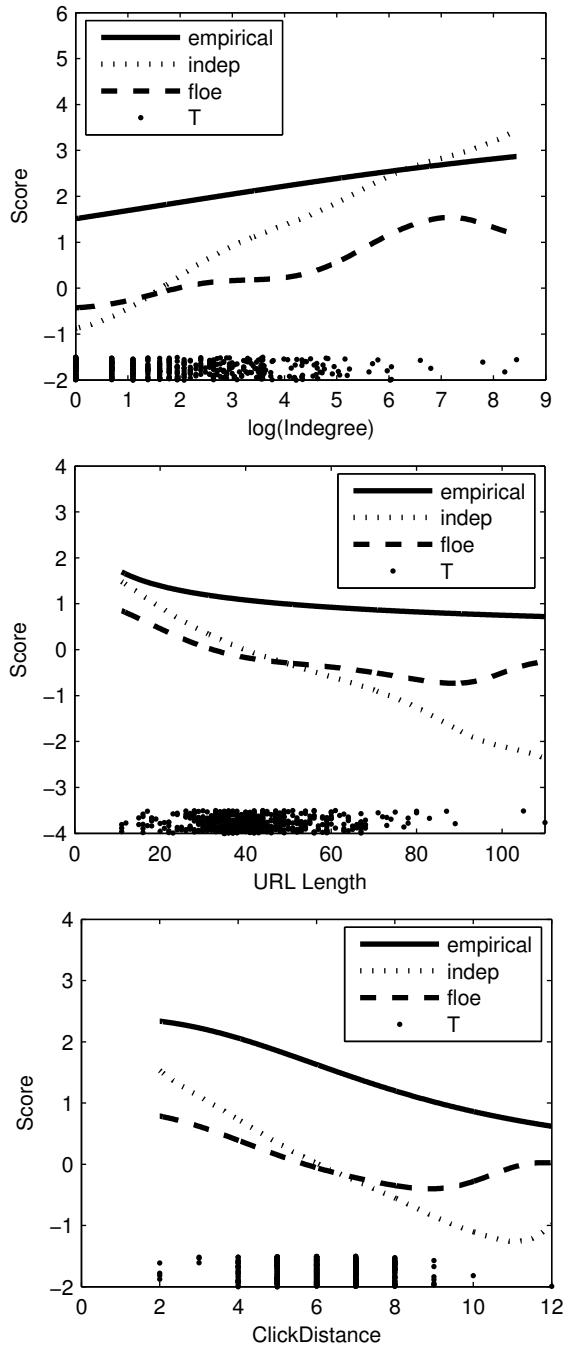

Figure 8: For our three new features, *floe* matches the best sigmoid score adjustment found empirically, whereas *indep* overstates the score adjustment (it is too steep).

set, since they are in some sense meant to uncover problems with our tuning before a test set is available.

Figure 9 shows that in most cases the *floe* line has become more uniform over S. For PageRank and indegree the line is very flat, indicating that no further adjustment is necessary. For URL length the line is non-zero in a region of sparse data, so perhaps does not indicate a systematic

Evidence	Combination	Training MAP	Test MAP
Baseline	—	0.458	0.430
PageRank	linear w=0.005	0.498	0.439
PageRank	log w=0.2	0.556	0.508
PageRank	w=1.34 k=1.36 (a=1)	0.556	0.515
PageRank	w=1.8 k=1 a=0.6	0.567	0.523
Indegree	w=3.6 k=5 a=0.2	0.549	0.489
URL Length	w=4.5 k=4 a=0.5	0.530	0.477
ClickDistance	w=2.4 k=8 a=2.6	0.532	0.470
PageRank + URL Length	w=1.8 k=1 a=0.6 w=1.9 k=6 a=0.2	0.572	0.532

Table 1: Static evidence performance on training and test sets. PageRank functions are linear, log, sigmoid with a=1 and full sigmoid. For indegree, URL Length and ClickDistance, we use full sigmoid only.

problem worth correcting. By contrast the ClickDistance line is distinctly non-uniform. We have so far been unable to find a function with better MAP performance than the sigmoid of ClickDistance. Either the plot is misleading, or there really is a better combination function we have not yet discovered. Note, we know it is possible for the ClickDistance line to be more uniform, for example, PageRank makes it so in Figure 11.

A limitation of this type of check is that, for our training set, we were not able to see differences between the PageRank weighting functions in Section 3.1. The differences in their effectiveness were too small to be reflected in the three corresponding post-check lines. This suggests that post-combination checking may be most useful for finding and understanding gross errors in combination, rather than deciding between functional forms which have similar performances. The level of noise in the estimates, at least on our training set, is too great to see finer differences.

To explore this, we plotted post-combination indegree lines for a range of w values (Figure 10). The numbers in brackets are the performance on the training set. Despite noise, the best-performing tunings do have the flattest lines. When there are gross errors in weighting, the slope of the line correctly indicates the type of further upweighting or downweighting required to correct the problem. We believe that when lines become flat in Figure 9 for PageRank, indegree and URL length, this indicates at least no gross errors of combination were made. More investigation is needed for ClickDistance.

Post-combination checking can also be used across features. For example, PageRank was the best single static feature (Table 1). Having added PageRank, does FLOE indicate that adjustment of other static features is now necessary?

Figure 11 shows *floe* lines for indegree, URL length and ClickDistance on top of a baseline of BM25 plus PageRank. The indications for indegree and ClickDistance are that no further adjustment is necessary. This makes sense since they are both link metrics. The lines are flat, at least in the region where we have sufficient data.

The *floe* line for URL length is above zero for very short URLs, which is the largest deviation at the left-hand (non-

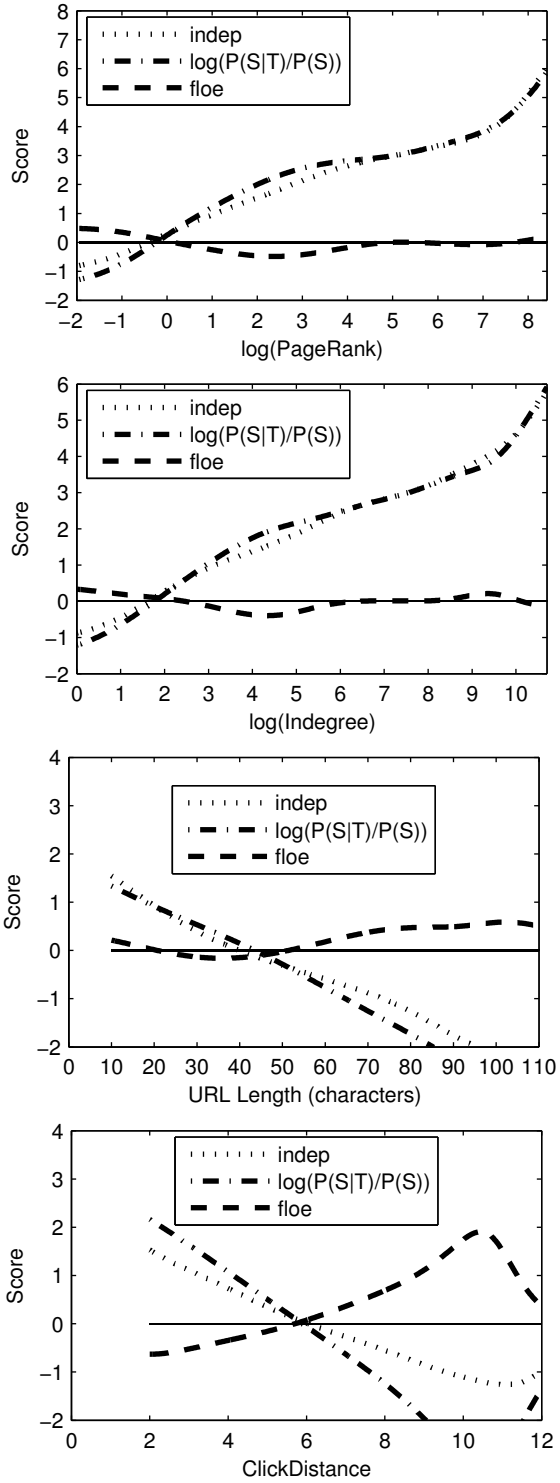

Figure 9: After adding a feature, the *floe* line tends to become flat for that feature, indicating that no further adjustment is necessary. ClickDistance is the exception.

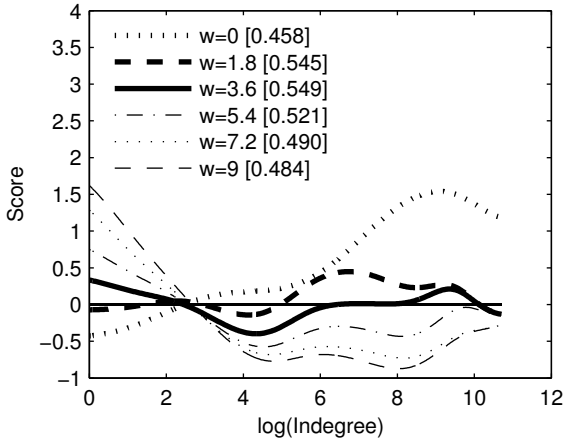

Figure 10: Post-correction *floe* lines for indegree. Overweighted indegree gives a downward slope, underweighted gives upward slope, with best-performing lines being flat.

sparse) region of the three plots. This indicates it may still be useful to positively weight short URLs. URL length is not a link metric, so it is plausible that it contains information that is not in PageRank.

To test these predictions we tuned each feature on top of our best BM25+PageRank baseline. The weight of indegree went to zero and the weight of ClickDistance went to 0.1, making an improvement in MAP at the fifth decimal place. Therefore we take it that there was no improvement to be had, at least by our combination methods. By contrast URL Length gave an improvement, as predicted by FLOE. The result is at the bottom of Table 1.

6. DISCUSSION AND CONCLUSION

Our starting position was that it is natural in a ranking system based on relevance weighting to add an extra weight for a query independent feature. In Section 3.2 we developed the correct relevance weight for use in a situation where we assume content match M and static feature S are independent (*indep*). However, when we estimate this weight it does not match weights we find empirically (Figures 5 and 8). Through a simple heuristic, adjusting for the levels of S already present in M, we develop a much more accurate predictor (FLOE).

We use this model as an indication of the appropriate functional forms for indegree, URL length and ClickDistance. We use it again as a test that we have made no gross errors in combination (suggesting we need to do more work on ClickDistance). Finally, we use it as a light-weight test, as an indication that on top of a BM25 and PageRank baseline, not much further adjustment is needed. In the scenario of an operational search system, correctly making such a prediction would allow a feature to be eliminated early, perhaps avoiding implementation effort.

There are two major limitations with this work. First, there is noise in the estimation, particularly when data becomes sparse. The level of noise present in the estimates in Figure 10 means we talk about what the line 'indicates' rather than taking its meaning as precise. The second limitation is related to the first. If the estimate were very good,

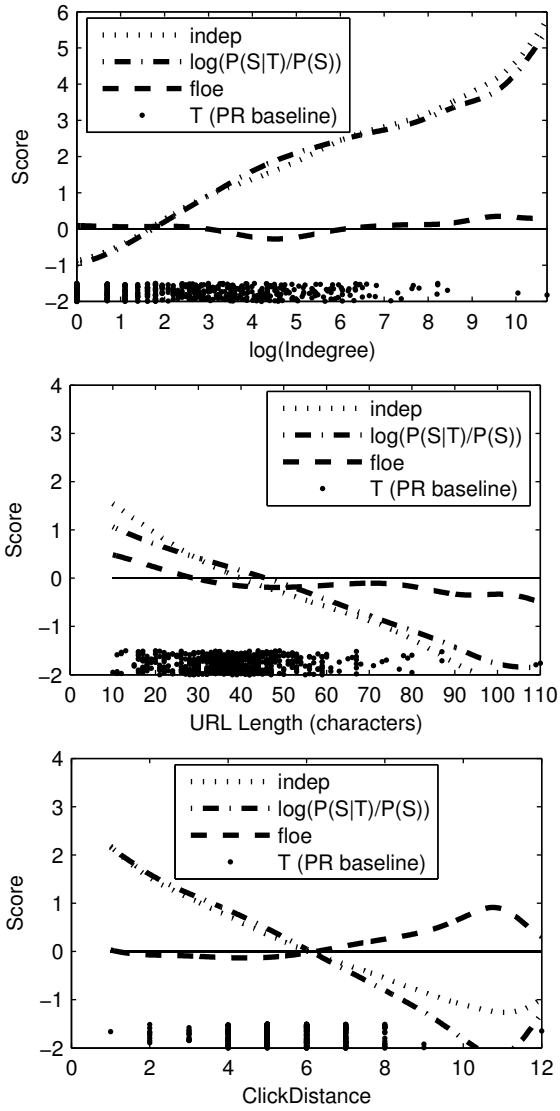

Figure 11: For a baseline of BM25 plus PageRank, the flat *floe* lines indicate not much adjustment is needed. There is an upward slope for short URLs.

it would be possible to skip the step of tuning on the training set and we could use the *floe* line itself as an adjustment. We have begun experiments in this direction, for example tuning a sigmoid directly to fit the estimate, but so far this gives inferior results.

One major outcome is an effective system for adding static features using relevance weights. Estimates for PageRank, link indegree, URL length and ClickDistance all indicate that a sigmoid functional form is appropriate, although more investigation is needed for ClickDistance. Each feature on its own can be used to improve effectiveness. However, after adding PageRank, the only additional static feature needed was URL length, and this gives only a slight improvement. So a very specific result of our experiments is a Web ranking scheme using field-weighted BM25 plus sigmoid-transformed PageRank and URL length.

FLOE has offered useful guidance in developing this system. We look forward to applying the model for new static features, perhaps in different settings. For example, there are several potentially interesting static features in email search, which might be explored in the planned TREC-2005 Enterprise Track.

Acknowledgements

Thanks to Marc Najork for calculating PageRank on .GOV.

7. REFERENCES

[1] B. Amento, L. Terveen, and W. Hill. Does "authority" mean quality? predicting expert quality ratings of web documents. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 296–303, New York, NY, USA, 2000. ACM Press.

[2] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, Oxford, UK, UK, 1996.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[4] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma. Block-level link analysis. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 440–447. ACM Press, 2004.

[5] N. Craswell and D. Hawking. Overview of the trec-2004 web track. In *Proceedings of TREC-2004*, Gaithersburg, Maryland USA, November 2004.

[6] R. Fagin, R. Kumar, K. S. McCurley, J. Novak, D. Sivakumar, J. A. Tomlin, and D. P. Williamson. Searching the workplace web. In *WWW '03: Proceedings of the twelfth international conference on World Wide Web*, pages 366–375. ACM Press, 2003.

[7] J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in xml retrieval. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 80–87. ACM Press, 2004.

[8] I.-H. Kang and G. Kim. Query type classification for web document retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 64–71. ACM Press, 2003.

[9] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 27–34. ACM Press, 2002.

[10] G. Pandurangan, P. Raghavan, and E. Upfal. Using pagerank to characterize web structure. In *COCOON '02: Proceedings of the 8th Annual International Conference on Computing and Combinatorics*, pages 330–339. Springer-Verlag, 2002.

[11] S. Robertson, H. Zaragoza, and M. 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. ACM Press, 2004.

[12] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM Press, 1996.

[13] T. Upstill. *Document ranking using web evidence*. PhD thesis, Australian National University, 2004.

[14] T. Upstill, N. Craswell, and D. Hawking. Query-independent evidence in home page finding. *ACM Trans. Inf. Syst.*, 21(3):286–313, 2003.

[15] H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC–13: Web and HARD tracks. In *Proceedings of TREC-2004*, Gaithersburg, Maryland USA, November 2004.