Seeing the forest from trees : Blog Retrieval by Aggregating Post Similarity Scores

Zhixin Zhou
School of CS & IT
RMIT University
VIC 3001, Australia
zhixin.zhou@rmit.edu.au

Xiuzhen Zhang
School of CS & IT
RMIT University
VIC 3001, Australia
xiuzhen.zhang@rmit.edu.au

Phil Vines
School of CS & IT
RMIT University
VIC 3001, Australia
phil.vines@rmit.edu.au

Abstract *Blog retrieval is a new and challenging task. Instead of retrieving individual documents, this task requires retrieving collections of documents, or blog posts. It has been shown recently that the federated model of using post entries as retrieval units is an effective approach to blog retrieval, where aggregation of similarity scores for posts to rank blogs plays an important role in the final ranking of blogs. In this paper, we explore two approaches of aggregation describing the depth and width of topical relevance relationship between post entries and blogs. We further propose holistic approaches that combine both approaches. Our experiments show that the sum baseline has the best performance, although the performances of the probabilistic approach and the linear pooling approach are very similar.*

Keywords blog retrieval, score aggregation

1 INTRODUCTION

The past decade has seen a surge of user-generated data on the web, among which the blogs play an important role. The term *blogosphere* refers to the whole collection of blogs on the Web.

A *blog* (also referred to as *web log*) is usually created and maintained by a web user who shares his or her writings on the web. Each *entry* of such writings is called a *blog post*, and is often followed by a list of replies from other web users, who contribute to the blog by adding their responses. Readers of blogs usually follow a subscription pattern, where they see an interesting blog post, browse through the other posts in the same blog, and if still interested, subscribe to the blog with a reader software which automatically tracks the updates through a file known as the *blog feed* (in the format of RSS[1] or ATOM[2]). One of the most popular instances of this kind of software is *Google Reader*[3].

[1]http://en.wikipedia.org/wiki/RSS
[2]http://en.wikipedia.org/wiki/ATOM
[3]http://reader.google.com

Proceedings of the 15th Australasian Document Computing Symposium, Melbourne, Australia, 4 December 2010.

A typical blog search behavior is shown in Figure 1, where *Jane Anderson* is a fictional blog author who frequently comments on brands of cosmetics. Many large companies also maintain official blogs for customer support or marketing purposes, among which is Google[4]. In the example given, a web user wishes to explore public opinions about Lancome, while another user would like to be informed about news from Google. After finding the blogs, the two users subscribe to them via feeds.

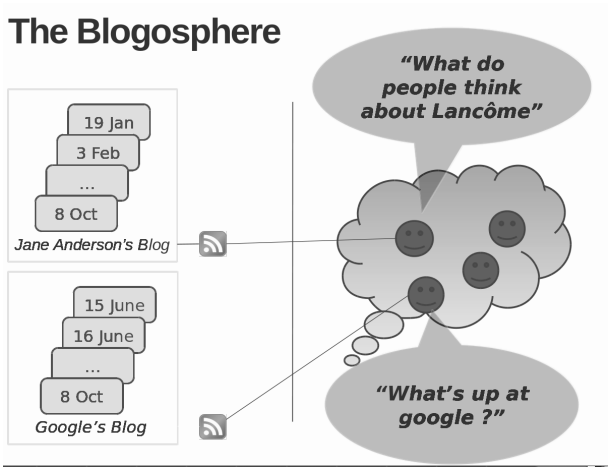

Figure 1: The blog searching behavior

Blog[5] retrieval, also known as *blog search* or *feed retrieval*, has many distinguishing characteristics from traditional document retrieval [5]. One of the essential differences is that the retrieval unit is no longer a single document, but a collection of documents (blog posts) to be evaluated as a whole. The TREC conference set up the Blog Track [13, 9, 12, 10] in 2007 to study search behaviors in the blogosphere, and introduced the Blog Distillation Task in 2007 to address the specific challenges posed by blog retrieval, also known as *blog feed search* [5]. The task is defined as to "*find me a blog with a principal, recurring interest in* X", where X is a topic of interest. Although a blog can be arguably viewed as a large virtual document comprising all posts to which

[4]http://googleblog.blogspot.com/
[5]In this paper, we refer to blogs and feeds interchangeably as there is a one-to-one mapping between the two.

traditional retrieval techniques can be directly applied, recent work by Elsas et.al [5] has shown that a federated model that considers the topical relationship between a blog and its post entries is an effective approach to blog retrieval.

The aggregation of similarity scores for posts to rank blogs to establish the topic relevance relationship between posts and blogs and therefore plays an important role in effective blog retrieval. In this paper we explore aggregation approaches for combining both views of topical relevance relationship between blogs and their posts. Our problem can be described as follows. Existing document retrieval systems are able to estimate query relevance at the post level, and often in the form of similarity scores. Assuming that such scores are a reasonable measure of the post-level relevance, our research question can be stated as: *What is the best approach to aggregate post similarity scores for blogs so as to rank the blog feeds by query relevance?*

First we need to define what a *relevant blog* is. However, there is not a widely adopted definition of a relevant blog. According to the TREC definition of the blog distillation task, blogs that show "*principal, recurring interest*" are the target of retrieval. In terms of the topic relevance relationship between blogs and their post entries this definition expresses the depth and width aspects of the topical relevance relationship between blogs and their post entries. In our discussions we use both sets of terms interchangeably.

Unfortunately, *principal* and *recurring* address different dimensions of the relevance. The *principal* dimension focuses on the relative percentage of relevant content in a blog, while the *recurring* dimension indicates the absolute amount of relevant content. Let us consider two blogs, one with 100 posts among which 20 were relevant, the other with 10 posts which are all relevant, and we assume that the relevant posts share the same degree of relevance. In this scenario the first blog shows more recurring interest as it has twice the number of relevant posts, whereas the second shows more principal interest since all of its posts were related to the topic.

A natural question is which one of the principal and recurring dimensions of relevance is more important for determining blog relevance to a topic. We study both aspects in terms of the topical relevance relationship between blogs and their posts. We also propose approaches combining the two aspects for ranking blogs.

The rest of this paper is organized as follows. In Section 2, we briefly survey related work on blog retrieval. In Section 3, we provide a detailed explanation of the approaches proposed. In Section 4, we present our framework for evaluation. In Section 5, we describe the setup of our experiments and discuss the results. Finally, the conclusion is made with an outlook on future works in Section 6.

2 RELATED WORK

The recent work of Elsas et.al [5] adapted a federated search model for blog retrieval. They showed that the federated model with blog posts as retrieval units outperformed the large-document model viewing a blog as a large documents comprising all posts. The focus of their work is on studying the pseudo-relevance feedback for posts and adapting it to improve blog retrieval. Our work is based on a similar blog retrieval model where post entries are the base retrieval units. We instead focus on how to aggregate the post relevance to achieve effective blog retrieval, which complements their study.

Blog search is a relatively new task. Related work started in 2007, when the blog distillation task was introduced into the TREC Blog Track [9, 12, 10]. Many participating groups approached this task by adapting techniques used in other existing search tasks. In this paper we focus only on major approaches used in post score aggregation, or in other words the document representation model.

He et al from the University of Glasgow [7, 6] compared the blog distillation task to the *expert finding* task of the Enterprise track [1]. Expert finding is the task of ranking candidate people as potential subject matter experts with respect to a given query [2].Each expert is associated with a collection of documents, and the retrieval model for this task assumes that expert candidates would have a large number of documents relevant to the query. This is similar to the blog retrieval task where each blog has a collection of blog posts, and a relevant blog would have a large number of relevant posts. He et.al adapted their Voting Model used in expert finding to feed search. Their model considers both the count of relevant posts in a feed and the extent to which each post is relevant.

Seo et al from the University of Massachusetts [15, 16] viewed this task as a *resource selection* problem, which originated from the distributed retrieval paradigm. Distributed information retrieval is also known as federated search, deals with document retrieval across a number of collections. The resource selection task aims to rank the document colletions so as to select the ones which are most likely to contain a large number of relevant documents. In their work, the geometric mean of the the query likelihoods of "pseudo-clusters", which are essentially the most relevant posts in a blog, was used to evaluate the blog's relevance to the query.

Our work is different from the above described approaches in many ways. First, we focus our study on score aggregation and do not use any query expansion module or proximity search techinques. Second, we examine the relevance of blogs in two dimensions, which have not been addressed this way by previous works. Third, we have proposed holistic score aggregation approaches combing the two dimensions of post relevance, which are both shown to be effective.

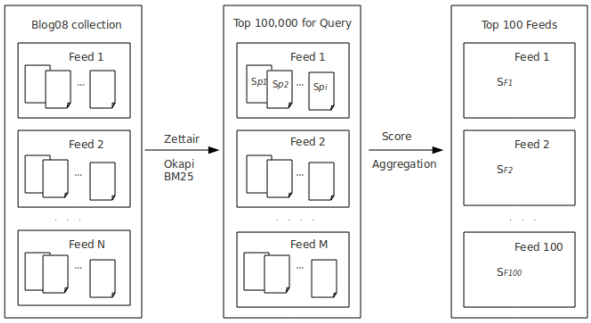

Figure 2: The Blog Retrieval Framework

3 THE BLOG RETRIEVAL FRAMEWORK

Our overall blog retrieval framework is illustrated in Figure 2. The framework comprises two components, namely, post retrieval and post aggregation. First we employ a document retrieval system to rank all blog posts in the collection by their estimated relevance to to a given query, and retrieve the top N. Ideally, to examine both the depth and width of relevance of a blog to a topic, the topical relevance of all post entries should be considered. However, given the size of the collection (the Blog08 collection we used has 28,488,766 unique posts), taking all posts into consideration is computationally costly. When N is sufficiently large, we could assume that the top N posts are a good representation of the collection with regard to the given topic, as the posts beyond this range are very unlikely to be relevant according to the estimation made by the retrieval system. To simulate the situation in our setting and also make the task manageable, for each topic, only the top 100,000 posts returned by the search engine were kept in the pool. In our experiments we also applied different thresholds to the post similarity scores, so that sub-collections with different levels of average query relevance are selected for investigation.

Theoretically any document retrieval model can be applied to retrieve blog posts. In our implementation we use the Zettair search engine [6] as the core document retrieval system. Zettair is a fast text search engine developed by the Search Engine Group at RMIT University. The Okapi BM25 model [8] is the core retrieval algorithm in Zettair, which is a probabilistic model. The similarity score of a document to a query, denoted as $S_{q,d}$, is an estimation of how closely the content of the document matches the query. The Okapi BM25 model makes use of statistical information about the distribution of the query terms (within both the document and the collection as a whole) to calculate the post similarity score.

$$S_{q,d} = \sum_{t \in \tau_{q,d}} w_t \times \frac{(k_1+1)f_{d,t}}{K + f_{d,t}} \times \frac{(k_3+1)f_{q,t}}{k_3 + f_{q,t}} \quad (1)$$

[6] http://www.seg.rmit.edu.au/zettair/

The parameters in the equation are shown below,

q The query

d The document (blog post)

t A term in the query

w_t A representation of the inverse document frequency, calculated by $w_t = \ln \frac{N_d - N_{d_t} + 0.5}{N_{d_t} + 0.5}$ where N_d is the number of documents in the whole collection and N_{d_t} is the number of documents that contains the term t

$\tau_{q,d}$ The intersection of the distinct terms from the query and the document

k_1 A constant within the range of 1.2 to 1.5

k_3 A constant set to 1000000 (effectively infinite)

b A constant within the range of 0.6 and 0.75

K Calculated by $K = k_1 \times \left[(1-b) + \frac{b \cdot W_d}{W_{AL}}\right]$, where W_d is the document length and W_{AL} is the average document length in the collection

The last step of our blog retrieval framework is aggregating the query relevance for posts to score and rank blogs for topical relevance. This is a crucial step for the effectiveness of the whole system. Most existing systems estimate the query relevance in the form of similarity scores.These scores may or may not be distributed in a uniform manner [11], but it is usually possible to transform them into a uniform space. We consider the width and depth dimensions of the topical relevance of blogs, in terms of aggregating post similarity scores. We first discuss baseline approaches for aggregation in the next section and then propose two holistic approaches later.

4 BASELINE APPROACHES TO SCORE AGGREGATION

Corresponding to the width and depth dimensions of topic relevance of blogs we propose two baseline approaches aggregating post similarity scores to produce the blog relevance score. Given a blog F of n posts and post relevance scores $\{s_1, s_2, ..., s_n\}$, the blog similarity score S_F could be calculated from the post relevance scores $\{s_1, s_2, ..., s_n\}$,

The Average Baseline With the average baseline approach, the average post similarity score is used to estimate the query relevance of a blog, as shown in the equation below.

$$S_F = \frac{\sum_{i=1}^{n} s_i}{n} \quad (2)$$

where s_i is the similarity score of the i^{th} post.

This approach addresses the aspect of "*principal interest*" as stated in the definition of the blog distillation task by TREC. It reveals the relationship between the

average degree of the query relevance of posts and that of the blog. Intuitively, blogs with a large number of posts are penalized. For instance, a blog F_A with 100 posts among which 10 were relevant should be considered more relevant than a blog F_B with 10 posts among which 5 were relevant. However, the average post relevance of F_A is likely to be lower than that of F_B.

The Sum Baseline With the sum baseline approach, the sum of post similarity scores is used to estimate the query relevance of a blog, as shown in the equation below.

$$S_F = \sum_{i=1}^{n} s_i \tag{3}$$

where s_i is the similarity score of the i^{th} post. This approach addresses the aspect of "*recurring interest*" as stated in the definition of the blog distillation task by TREC. It reveals the relationship between the absolute amount of relevant content in a blog and its overall relevance to the blog. Intuitively, it discriminates against blogs that are specialized in one topic but having a low count of posts.

5 THE PROBABILISTIC MODEL

We propose to estimate the likeilihood of a blog's relevance to a given query from the degree of relevance of its post entries. In this approach we assume that a blog is considered irrelevant only if all posts in the blog are irrelevant. To calcluate the probability of the event that a blog be relevant to the query, we first transformed the post similarity scores in a feed F into probabilistic values,

$$p_i = \frac{s_i - s_{Q_{lower}}}{s_{Q_{upper}} - s_{Q_{lower}}} \tag{4}$$

where p_i is the probability of the i^{th} post being relevant to the query, Q_{upper} is the highest similarity score of all posts relevant to the query Q, Q_{lower} is the lowest similarity score of all posts relevant to the query Q, and s_i is the similarity score of the i^{th} post. We performed the transformation on a per-topic basis, as we expected different distributions of post similarity scores for each topic. Based on our assumption, the probability of the blog being irrelevant can then be calculated as,

$$\bar{P}_F = \prod_{i=1}^{n}(1 - p_i) \tag{5}$$

As a blog can be either relevant or irrelevant, the probability of a blog being relevant is thus,

$$P_F = 1 - \prod_{i=1}^{n}(1 - p_i) \tag{6}$$

Intuitively, this approach would not work well with blogs with a large count of irrelevant posts, as $\prod_{i=1}^{n}(1 - p_i)$ shrinks dramatically even if p_i is sufficiently small. We circumvent this problem by applying a threshold on p_i, so that only the "relevant" posts are considered. Here, the probability of the blog being irrelevant is no longer the probability of all its posts being irrelevant. Instead, it is calculated as the probability of all "relevant" posts in this feed being irrelevant, where the "relevant" posts are selected by the threshold applied on the similarity score of the posts. Effectively this is setting the probability of low-score posts being relevant to zero. And since the similariy score is a reasonable indicator of the query relevance, the low-scored posts can be assumed to be irrelevant.

6 LINEAR POOLING: A HOLISTIC APPROACH

We propose an approach combining the depth and width dimensions of topical relevance for aggregating post similarity scores. As will be discussed in the Experiments section, the approach outperforms the baseline approaches.

Pooling distributions is a general approach to combining information from multiple sources or approaches, where sources are typically represented as probability distributions [4]. Here we consider two approaches, one based on the average baseline model and focusing on the width of relevance and the other based on the sum baseline model and focusing on the depth of relevance. We adopt the linear pooling approach to aggregating the estimations from these two approaches.

The distributions of scores over the two baselines are different. The scores from the average baseline range from 0 to 1, while those from the sum baseline can value above 100. Therefore we transform the feed scores into z-scores first, and combine the z-score value for the blogs. The z-score is calculated by the following formula:

$$s_Z = \frac{s - \mu}{\sigma} \tag{7}$$

where μ is the mean of all scores in the current distribution, and σ the standard deviation. Since the z-score measures the distance between a score and the mean score in the distribution and normalize that with the standard deviation, it allows scores form two different distributions to be comparable to each other.

We combine the two scores for each feed as follows:

$$P_F = \alpha * S_{F1} + (1 - \alpha) * S_{F2} \tag{8}$$

where S_{F1} and S_{F2} are the z-scores computed from the two probability values for blog relevance, calculated by Equation 4 for the average baseline model and the sum baseline model. Note that α and $1 - \alpha$ are the weights for the average baseline model and the sum baseline model respectively. By default we set $\alpha = 0.5$. Generally α can be adjusted to bias towards the depth dimension or the width dimension.

Table 1: Blog08 Collection Statistics (sourced from TREC Overview Paper[10])

Quantity	Blog08
Number of Unique Blogs	1,303,520
First Feed Crawl	14/01/2008
Last Feed Crawl	10/02/2009
Number of Permalinks	28,488,766
Total Compressed Size	453GB
Total Uncompressed Size	2309GB
Feeds (Uncompressed)	808GB
Permalinks (Uncompressed)	1445GB
Homepages (Uncompressed)	56GB

7 EXPERIMENTS

7.1 Dataset

Our experiments were done on the Blog08 collection used for TREC 2009 and TREC 2010 Blog Track conferences. This collection was created by the University of Glasgow to provide an experimental environment for the Blog Track. Summary statistics for this collection are listed in Table 1.

The collection contains three types of data, namely, permalinks (blog posts), feeds, and homepages. We only indexed the permalinks in our experiments. Each permalink document is associated wth one feed, whereas a feed could be associated with multiple permalink documents. On average, each blog contains 22 posts in our collection.

We tested our approaches with the 49 topics used in TREC 2009 Blog Track. We used only the topic title text as our queries, which are typically short expressions comprising of two or three words such as "genealogical sources" (topic 1101). The query relevance judgments were done by NIST, against which the estimations made by our blog retrieval system were evaluated.

7.2 Evaluation

We follow the evaluation methods adopted by the Text REtrieval Conference [14, 3]. Four metrics were used, namely, MAP, P@10, R-prec, and B-pref. Each of these metrics address a different aspect of the performance of the retrieval system, where the MAP is the main measure of the system's performance. This is also the main measure used in the TREC Blog Track. The results we show are generated by the trec_eval [7] software provided by the TREC conference.

MAP Precision and recall are single-value metrics based on the whole list of documents returned by the system. For systems that return a ranked sequence of documents, it is desirable to also consider the order in which the returned documents are presented. MAP, or Mean Average Precision, emphasizes ranking relevant documents higher. It is the average of precisions computed at the point of each of the relevant documents in the ranked sequence.

P@10 P@10 (Precision at 10 documents) counts the number of relevant documents in the top 10 documents in the ranked list returned for a topic. This precision correlates with the precision observed by a web user.

R-prec R-prec is the precision computed after R documents have been retrieved, where R is the number of relevant documents for the topic. Contrary to MAP, this metric de-emphasizes the exact ranking of the retrieved relevant documents.

B-pref B-pref is robust in collections which may have incomplete relevance information. The idea is to measure the effectiveness of a system on the basis of judged documents only. R-precision, MAP, and P(10) are completely determined by the ranks of the relevant documents in the result set, and make no distinction in pooled collections between documents that are explicitly judged as nonrelevant and documents that are assumed to be nonrelevant because they are unjudged. B-pref, on the other hand, is a function of the number of times judged non-relevant documents are retrieved before relevant documents.

We applied thresholds on the similarity scores to select different collections of posts with different levels of average relevance. First we scaled the similarity score of each post to a probability value between [0,1], with the method defined in Equation 4. Different threshold settings on these probability scores allow us to examine the performance of our approaches in different post collections in terms of average query relevance. For each topic, we applied 9 thresholds, namely 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, to each approach except the linear pooling approach. This is because with that approach, a majority of the topics do not have any post with a score above 0.8.

7.3 Results and Discussion

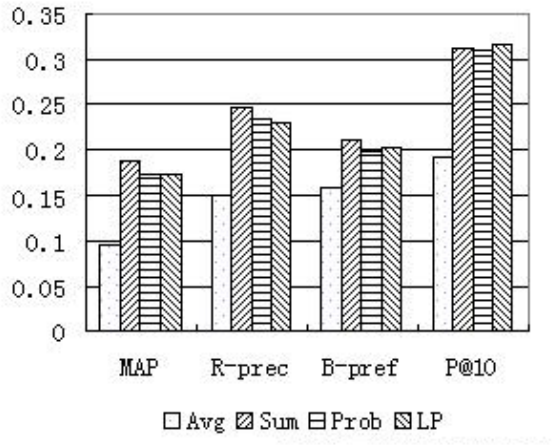

Figure 3: Overview

[7] http://trec.nist.gov/trec_eval/index.html

Figure 3 provides an overview of the approaches we used. The run label *Avg* corresponds to the average baseline, label *Sum* refers to the sum baseline, label *Prob* is for the probabilistic approach, and label *LP* refers to the linear pooling approach. As is shown in the graph, the sum baseline outperformed all other approaches when evaluated with all metrics used except P@10. The probabilistic approach and the linear pooling approach have similar performance to that of the sum baseline, but the average baseline is significantly worse than the other approaches. Note that all the data shown in this graph was not obtained under the same threshold setting. Instead, the best run for each approach was selected from a number of runs under different settings. The effect of different thresholds is discussed below.

We extracted individual topic performance by MAP for each approach, and used the paired Wilcoxon test to compare the difference between the approaches. The performance of the sum baseline, the probabilistic approach and the linear pooling approach were significantly better than that of the average baseline, with $p < 1.449e-6$, $p < 9.942e-5$, $p < 4.7e-6$ respectively. The performance of the sum baseline is significantly better than the linear pooling approach as well, with $p < 0.005184$, but the difference between its performance and that of the probabilistic approach is insignificant. There is no significant difference between the performance of the probabilistic approach and the linear pooling approach, either.

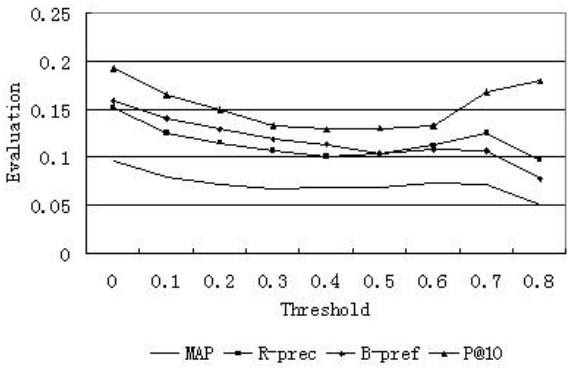

Figure 4: The Average Baseline

Figure 4 shows the performance of the average baseline with different thresholds applied to the post scores. With metrics other than b-pref, the performance deterioriates as the threshold gets higher, but improves slightly near the threshold 0.7. Above that threshold, the performance again declines with all metrics but P@10. This is proabably due to the fact that only a small number of blogs have posts with such a high score, and with such highly relevant posts they are very likely to be relevant. With P@10, only the top 10 blogs are evaluated, and is not affected by the decrease in the number of feeds found. The improvement over the performance reflected by P@10 also implies that highly relevant posts suggests a high blog relevance, but there is a tradeoff between the precision and the recall, as is reflected by the other metrics.

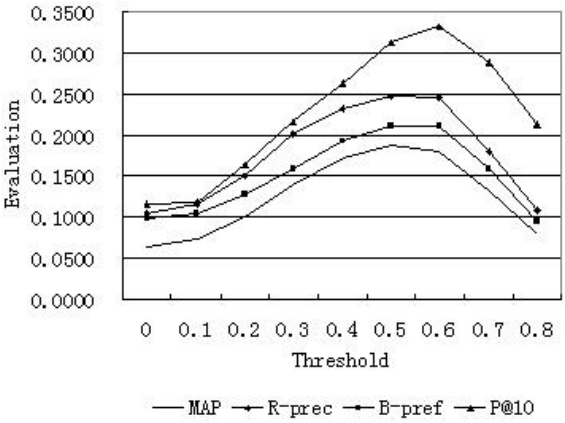

Figure 5: The Sum Baseline

Figure 5 shows the performance of the sum baseline under different thresholds. The trend shown in the graph is consistent with all metrics we used. The performance of the sum baseline improves as the thresholds becomes higher, and peaks near 0.5 and 0.6. After that, the performance declines, due to a decrease in the number of feeds found. This is in accordance with our observation with the average baseline. It also supports the implication that a group of highly relevant posts in a blog determines the query relevance of the blog.

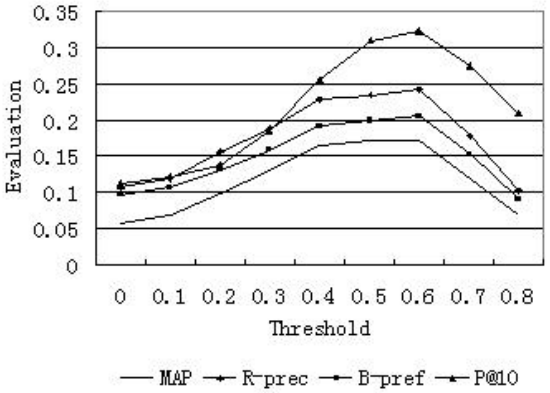

Figure 6: The Probabilistic Approach

Figure 6 shows the performance of the probabilistic model we proposed. Overall, the performance of this approach is very similar to that of the sum baseline. When evaluated with MAP, the performance of the sum approach peaked at 0.5, while the performance of the probabilistic model peaked at 0.6, but the difference between their performance at 0.5 and 0.6 was negligible.

Figure 7 and 8 shows the performance of the linear pooling approach we proposed. As is shown in Figure 7, the performance of this approach is similar to that of the sum baseline and the probabilistic model. It is worth noting however, that this approach has achieved the best

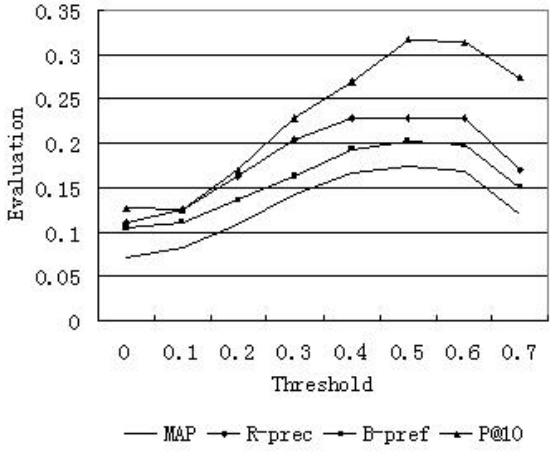

Figure 7: The Linear Pooling Approach

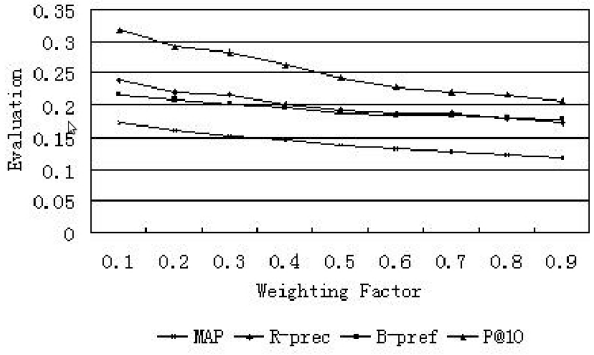

Figure 8: The Weigthing Factor α

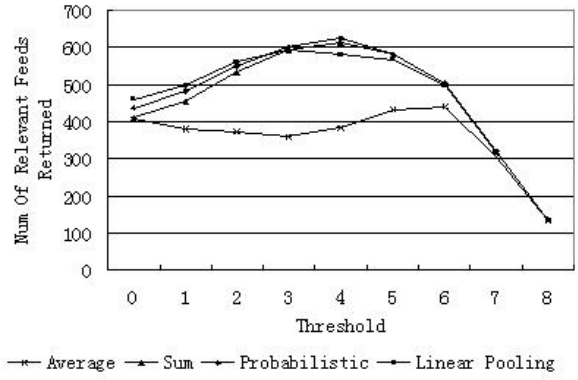

Figure 9: Num of judged relevant feeds returned

performance among all approaches when evaluated by P@10. Figure 8 shows how the weighting factor α (defined in Equation 8) influences the performance of the linear approach. In this graph we aggregated two baselines with different values for α. The two baselines we used were the average baseline with thresholds set to 0, and the sum baseline with thresholds set to 0.5. The choice of the thresholds was based on the performance of the two baselines we observed in our experiments, and we chose the ones which lead to the best evaluation results. However, as we have not tested the values exhaustively, the values we chose may not be optimal. From the graph we can see that the smaller the weighting factor is, the better performance appears to be. As a smaller α suggests larger weight for the sum baseline, and the best performance observed when $\alpha = 0.1$ is still inferior to that of the sum baseline, the graph implies that the average amount of the post relevance, or *the principal interest*, in a feed may not be as important as the total amount of the post relevance (taking only posts with a relatively high relevance score into account), or the *recurring interest*.

Overall, with all approaches other than the average baseline, the performance of the system is positively influenced by higher threshold settings, although it will be hurt by a drop in recall when the threshold rises to above 0.6, which is shown in 9. This observation implies that the posts with a high query relevance in a feed determines the blog relevance.

Interestingly, the performance of the average baseline actually declines with higher thresholds, although it rose a little when we used only posts with a score above 0.6. It is also worth noting that when using all posts (when threshold is 0), the average baseline has the best performance. Combined with our implication from the other three approaches, we deduce that the weights that the two aspect of the blog relevance carry vary under different threshold settings. While the average post relevance is a reasonable indicator of the blog relevance, the potentially huge amount of irrelevant posts could greatly hurt its viability. This also explains why the sum baseline has an extremely poor performance when using all posts (the accumulation of the low scores favors blogs with a large amount of posts). On the other hand, when considering two blogs A and B, both containing some highly relevant posts, and assuming that blog A has a larger count of highly relevant posts whereas blog B has a higher average post relevance, blog A is probably more relevant than blog B. This explains why elevating the threshold hurts the performance of the average baseline.

8 CONCLUSIONS AND FUTURE WORK

In this paper we have explored a number of approaches to estimate the query relevance of blogs. First we examined two baseline approaches based on the definition of the blog retrieval task, each addressing a different dimension of the blog relevance, namely, *principal* and *recurring*. Our experiment results show that highly relevant posts are a good representation of the feed in terms of its topical relevance. Our result also suggests that despite the fact that relevance judgment is affected by both the average degree of the query similarity and the number of relevant retrieval units, it is not sensitive to the former so long as a threshold is met.

We also proposed two holistic approaches which address both dimensions of the blog relevance. The two approaches have similar performances, both better than

the average baseline but very close to that of the sum baseline.

This work focused only on the score aggregation techniques. To further improve the performance of the approaches we proposed, we also plan to examine the techniques for score normalization. More flexible settings of thresholds based on the distribution of post scores within a blog are also subject to further study.

References

[1] P. Bailey, N. Craswell, A. P de Vries and I. Soboroff. Overview of the TREC 2007 enterprise track. In *Proceedings of TREC*, 2007.

[2] K. Balog, L. Azzopardi and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, page 4350, 2006.

[3] C. Buckley and E. M Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual internationa ACM SIGIR conference on Research and development in information retrieval*, 2004.

[4] R. T Clemen and R. L Winkler. Combining probability distributions from experts in risk analysis. *Risk Analysis*, Volume 19, Number 2, pages 187203, 1999.

[5] J. L Elsas, J. Arguello, J. Callan and J. G Carbonell. Retrieval and feedback models for blog feed search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, page 347354, 2008.

[6] D. Hannah, C. Macdonald, J. Peng, B. He and I. Ounis. University of glasgow at TREC 2007: Experiments in blog and enterprise tracks with terrier. In *Proceedings of TREC*, Volume 2008, 2007.

[7] B. He, C. Macdonald, I. Ounis, J. Peng, R. L Santos and GLASGOW UNIV (UNITED KINGDOM). University of glasgow at TREC 2008: experiments in blog, enterprise, and relevance feedback tracks with terrier. In *Proceedings of TREC*, 2008.

[8] K. Sparck Jones, S. Walker and S. E Robertson. A probabilistic model of information retrieval: development and comparative experiments:: Part 2. *Information Processing & Management*, Volume 36, Number 6, pages 809840, 2000.

[9] C. Macdonald, I. Ounis and I. Soboroff. Overview of the TREC 2007 blog track. In *Proceedings of TREC 2007*, 2007.

[10] C. Macdonald, I. Ounis and I. Soboroff. Overview of the TREC 2009 blog track. In *Proceedings of TREC 2009*, 2010.

[11] R. Manmatha, T. Rath and F. Feng. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, page 267275, 2001.

[12] I. Ounis, C. Macdonald, I. Soboroff and GLASGOW UNIV (UNITED KINGDOM). Overview of the TREC 2008 blog track. In *Overview of the TREC 2008 blog track*, 2008.

[13] I. Ounis, M. De Rijke, C. Macdonald, G. Mishne and I. Soboroff. Overview of the TREC-2006 blog track. In *Proceedings of TREC*, Volume 6, 2006.

[14] T. Sakai. Comparing metrics across trec and ntcir: the robustness to system bias. In *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008.

[15] J. Seo and W. B Croft. Umass at trec 2007 blog distillation task. In *Proc. of the 2007 Text Retrieval Conf*, 2007.

[16] J. Seo and W. B Croft. Blog site search using resource selection. In *Proceeding of the 17th ACM conference on Information and knowledge management*, page 10531062, 2008.