University of Lugano at TREC 2008 Blog Track

Shima Gerani, Mostafa Keikha, Mark Carman,
Robert Gwadera, Davide Taibi* and Fabio Crestani
University of Lugano
Department of Informatics
Lugano, Switzerland
{shima.gerani, mostafa.keikha, mark.carman, robert.gwadera}@lu.unisi.ch,
davide.taibi@uninsubria.it, fabio.crestani@unisi.ch

ABSTRACT

We report on the University of Lugano's participation in the Blog track of TREC 2008. In particular we describe our system for performing opinion retrieval and blog distillation.

1. INTRODUCTION

The 2008 Blog track continued on from the successful 2007 Blog track [12], including the same opinion retrieval and blog distillation activities. This year was our first participation in TREC and we participated in both opinion retrieval and blog distillation tasks. We aimed to test the effectiveness of learning methods in each of these tasks. In the topic retrieval phase (baseline) of the opinion retrieval task, we used a rank learning method [18] to combine additional information including the content of incoming hyperlinks and tag data from social bookmarking websites with our basic retrieval method which was the Divergence from Randomness version of BM25 (DFR_BM25) [2]. The results shows 20% improvement in the Mean Average Precision (MAP) of the proposed method in comparison with DFR_BM25. We then examined the effectiveness of learning methods in assigning opinion scores to documents. We compared a Support Vector Machine (SVM) based learning system with a simpler system that used the average opinionatedness of each word in the document. Although the results were not satisfactory in our TREC submission and we didn't improve the baseline, repeating the experiments showed the improvement over the baseline by using the learning methods in document opinion scoring.

In the distillation task we try to make use of link information for blog distillation. We first implemented a simple blog search method based on the voting model used in the expert search problem [11]. This baseline seems to perform better than the median of the TREC'08 participants. We then tried to take into account structure-based evidence using a rank learning approach. Unfortunately, the rank learning model appears to be very sensitive to properties of the data set, and did not perform well in our experiments.

The remainder of this paper is structured as fallows. We describe the method used in our baseline submission in section 2. In section 3 we discuss about our approach in opinion retrieval task. The distillation task is discussed in detail in section 4.

2. BASELINE RELEVANCE RETRIEVAL

For the baseline blog post retrieval task, we built a system that used a rank learning framework to combine relevance scores for each post with other forms of evidence. We first used the Terrier Information Retrieval system [14] to index the blog post (permalink) collection. We then tested various state-of-the-art retrieval models including BM25 [15], Divergence from Randomness [3] and Language Modeling [19], for generating relevance scores for individual posts. The TREC 2007 relevance assessments were used for the evaluation. The result of our analysis was that DFR_BM25 [2] produced the best result, with Mean Average Precision (MAP) of 0.2138.

We then extend this content-based retrieval technique with additional information including the content of incoming hyperlinks and tag data from social bookmarking websites. The latter has recently been shown to be useful for improving Web Search [9, 4, 17]. In order to incorporate these additional sources of evidence we rely on a rank learning approach [18, 5].

We trained an SVM-map [18] rank learner to optimally combine different forms of evidence into a single retrieval function. The different forms of investigated evidence were:

- the post content relevance score
- the inlink count, i.e. the number of incoming hyperlinks form other permalinks in the collection
- the cosine similarity between inlink anchor-text and the query
- the query length
- the popularity of the domain[1] (hostname) of the URL of the post on the social bookmarking (tagging) website Delicious[2]

[*] Visiting from the University of Insubria, Como, Italy.

[1] The domain of the permalink URL was used instead of the URL itself because of the sparsity of data on Delicious.

[2] http://delicious.com

Run	MAP	R-Precision	P@10
DFR_BM25	0.2138	0.3836	0.4087
run0	0.2663	0.2780	0.4273

Table 1: Topic-Relevance results for submitted baseline.

- a tag-query similarity score based on the relative frequency with which query terms are used to annotate the permalink's domain on Delicious.

Table 1 shows the topic relevance results of the DFR_BM25 method and our baseline system (run0). The results indicate that using a rank learning approach to include additional information such as the content of incoming hyperlinks and tag data from social bookmarking websites, can boost the performance of a baseline content-based retrieval system.

3. OPINION RETRIEVAL

Our approach to ranking blog posts by their opinionatedness again relies on a learning framework. In this case we train a Learning to Rank system to take both a relevance score (output by the rank learner described above) and an "opinion score" for each document into account when producing an output ranking. The advantage of this approach is that we do not need to decide explicitly how best to combine these forms of evidence, but can rely on historical data for fine tuning the retrieval.

The problem then is to estimate a score for the "opinionatedness" of each document. We have two approaches to doing this. In the first approach we calculate an opinion score for each term in the document and then combine the score over all terms in the document. In the second we train a classification system to distinguish between opinionated and non-opinionated posts. We then use the confidence of the classifier as an opinion score for the document.

To better describe these techniques, we introduce some notation. Assume that we have a set of labeled training documents, denoted D, and a set of training queries, denoted $q_1, ..., q_n$. For each query q_i we have a set of relevant documents $R_i \subset D$, and a set of opinionated documents $O_i \subset R_i$, that were judged by assessors to be relevant and opinionated respectively. Let $O = \cup_i O_i$ be the set of all opinionated documents in our training set and $R = \cup_i R_i$ be the set of all relevant documents, (note that $O \subset R$). The relative frequency of a particular term t in the set O is denoted $p(t|O)$ and calculated as:

$$p(t|O) = \frac{\sum_{d \in O} c(t,d)}{\sum_{d \in O} |d|} \tag{1}$$

where $c(t,d)$ denotes the number of occurrences term t in document d and $|d|$ denotes the total number of words in the document. We can now calculate an opinion score for each term t using the technique proposed by Amati et al. [1] as follows:

$$opinion(t) = p(t|O) \log \frac{p(t|O)}{p(t|R)} \tag{2}$$

Note that the summation over all terms of the opinion score gives the well-known Kullback-Leibler divergence [13] between the opinionated document set and the relevant document set. This measure quantifies the dissimilarity between the two sets of documents. Terms which cause high divergence are therefore good indicators of opinionatedness. In order to calculate an opinion score for an entire document, we can simply calculate the expected opinionatedness of words in document:

Run	MAP	R-Precision	bPref	P@10
OurBaseline	0.2663	0.3478	0.3799	0.4273
opin0kl	0.2080	0.2597	0.3284	0.4040
opin0svm	0.2075	0.2698	0.3325	0.3547
Baseline1	0.3701	0.4156	0.4501	0.7307
opin1kl	0.3662	0.4136	0.4468	0.7187
opin1svm	0.2936	0.3569	0.4102	0.4947

Table 2: Topic-Relevance results for submitted runs.

Run	MAP	R-Precision	bPref	P@10
OurBaseline	0.1902	0.2429	0.2566	0.2920
opin0kl	0.1525	0.1986	0.2263	0.2967
opin0svm	0.1797	0.2372	0.2756	0.2873
Baseline1	0.2639	0.3189	0.3170	0.4753
opin1kl	0.2626	0.3178	0.3144	0.4760
opin1svm	0.2511	0.3136	0.3312	0.4100

Table 3: Opinion Retrieval results for submitted runs.

$$opinion1(d) = \sum_{t \in d} opinion(t) p(t|d) \tag{3}$$

where $p(t|d)$ is the relative frequency of term t in d.

Alternatively, as stated previously, we can train a classification system and in particular a Support Vector Machine (SVM) to recognize opinionated documents. We can then use the confidence of the classifier (i.e. the distance from the hyperplane) as the opinion score for each document. The per term opinion score is used in this case only for feature selection, with only the top 1,000 "most opinionated" terms being used as features for the classifier:

$$opinion2(d) = f_{SVM}(\langle p(t_1|d), ..., p(t_m|d) \rangle) \tag{4}$$

where the function $f_{SVM}()$ is the output (confidence) of the trained SVM for a particular document and m is the size of the feature set (vocabulary).

For our TREC 2008 participation we trained a Learning to Rank system to rank results by combining our relevance score for the document with one of the two different opinion scores described above. Table 3 shows the opinion retrieval results of the two proposed methods using our baseline(opin0kl, opin0svm) and one of the TREC 2008 baselines (opin1kl, opin1svm). Opin0kl is the run in which we used our own baseline to find the relevant set of documents. In order to calculate the opinion score for documents we used the expected opinionatedness of words in document as described before. In opin0svm we also used our baseline, but we used the confidence of the trained SVM as an opinion score for documents. Opin1kl is the run in which we used the TREC baseline1 to get the list of relevant documents and then we used the expected opinionatedness approach to find the opinion score of the documents. Opin1svm used TREC baseline1 and the SVM score approach. Although we couldn't improve the baselines, comparing opin0kl and opin0svm, shows that using the SVM confidence as an opinion score works better than the expected opinionatedness approach on our baseline, but the reverse is true for the TREC baseline. After the TREC submissions, we repeated the experiments performing a more comprehensive study and using 10-fold cross-validation. We discovered that even for the TREC baselines, using an SVM to calculate opinion scores for documents works better than the expected opinionatedness approach. The results showed about 17% improvement over TREC baseline1.

A distinct advantage of our "multiple levels of learning" approach is that we maximize the use of available training data and thus do not need to rely on external sources of opinion-bearing word lists, that may not be well-suited to the blog opinion retrieval task.

4. BLOG DISTILLATION

Blog search users often wish to identify blogs about a given topic so that they can subscribe to them and read them on a regular basis [12]. The blog distillation task can be defined as: "Find me a blog with a principle, recurring interest in the topic X." Systems should suggest feeds that are principally devoted to the topic over the timespan of the feed, and would be recommended to a user as an interesting feed about the topic (i.e a user may be interested in adding it to their RSS reader).

The blog distillation task has been approached from many different points of view. In [6], the authors view it as ad-hoc search and consider each blog as a long document created by concatenating all postings together. Other researchers treat it as the resource ranking problem in federated search [7]. They view the blog search problem as the task of ranking collections of blog posts rather than single documents. A similar approach has been used in [16], where they again consider a blog as a collection of postings and use resource selection approaches. Their intuition is that finding relevant blogs is similar to finding relevant collections in a distributed search environment. In [12], the authors modelled blog distillation as an expert search problem and use a voting model for tackling it.

Our intuition is that each posting in a blog provides evidence regarding the relevancy of that blog to a specific topic. Blogs with more (positive) evidence are more likely to be relevant. Moreover, each posting has many different features like content, in-links, and anchor text that can be used to estimate relevancy. There are also global features of each blog like the total number of postings, the number of postings that are relevant to the topic and the cohesiveness of the blog that could be useful to consider. The next sections are organized as follows. First we describe our approach. Then we show our experimental results and after we discuss about conclusion and future works.

5. APPROACH

Our first approach is to create a baseline for blog distillation system that uses only the content of blog posts as a source of evidence. To do this, we consider the expert search idea proposed in [8]. The main idea of that work is to treat blogs as experts and feed distillation as expert search. In the expert search task, systems are asked to rank candidate experts with respect to their predicted expertise about a query, using documentary evidence of expertise found in the collection. So the idea is that the blog distillation task can be seen as a voting process: A blogger with an interest in a topic would send a post regularly about the topic, and these blog posts would be retrieved in response to the query. Each time a blog post is retrieved, it can be seen as a vote for that blog as being relevant to (an expert in) the topic area.

We use the voting model to find relevant blogs. The model ranks blogs by considering the sum of the exponential of the relevance scores of the postings associated with each blog. The model is one of the data fusion models which Macdonald and Ounis used in their expert search system[11].

$$score\_cand(B,Q) \;=\frac{1}{|B|}\sum_{p\in R(Q)\cap B} exp(score(p,Q)) \qquad (5)$$

where $R(Q)$ is the set of retrieved postings for the query Q, and $|B|$ is the set of posts for blog B.

For our second approach, we investigated= using more features to represent each blog beside its content. To take the different features into account, we use a Rank Learning [18] approach to combine the features into a single retrieval function. Features that we thought could be useful were:

- Cohesiveness of blog postings
- Number of postings
- Number of relevant postings (posts in top N relevance results)
- Number of inlinks
- Relevance of inlink post content
- Relevance of inlink anchor-text

Table 5 shows features that were used to learn a retrieval function. In the formulas, $|B|$ denotes the number of posts in the blog B, *inlinks-source*(B) denotes the set of postings that contain a link to (a post in) blog B, *inlinks-anchor-text*(B) is the set of anchor-texts for those links, and $KL(p||B)$ is the Kullback-Leibler divergence between a posting p and its blog B:

$$D_{\mathrm{KL}}(p\|B) = \sum_{t\in p} w_p(t)\log\frac{w_p(t)}{w_B(t)} \qquad (6)$$

where $w_p(t)$ and $w_B(t)$ are the relative frequency of the the term t in the post and blog (as a whole) respectively.

6. EXPERIMENTAL RESULTS

For evaluating our methods we used the TREC Blogs06 test collection, which is a crawl of 100k blogs over an 11-week period [10]. This dataset includes the blog postings (permalinks), feeds and homepages for each blog. In our experiments we use only the permalinks component of the collection, which consists of approximately 3.2 million documents [8]. We used the Terrier Information Retrieval system[3] to index the collection.

To find relevant posts for each topic, the DFR_BM25 weighting model was used to compute a score for each blog post [14]. We chose this weighting model based on its performance on TREC'06 and '07 blog post opinion retrieval

[3] http://ir.dcs.gla.ac.uk/terrier/

Feature Name	Description
Blog Relevancy	$\sum_{p \in R(Q) \cap B} exp(score(p, Q))$
Cohesiveness	$\sum_{p \in B} KL(p\|\|B) \div \|B\|$
Normalized Blog Relevancy	Blog Relevancy $\div \|B\|$
Fraction of Relevant Postings	$\|B \cap R(Q)\| \div \|B\|$
Inlink Content Relevancy	$\sum_{d \in inlinks\text{-}source(B)} exp(score(d, Q)) \div \|inlinks(B)\|$
Inlink Anchor-Text Relevancy	$\sum_{a \in inlinks\text{-}anchor\text{-}text(B)} exp(score(a, Q)) \div \|inlinks(B)\|$

Table 4: Selected features for learning phase of blog distillation approach.

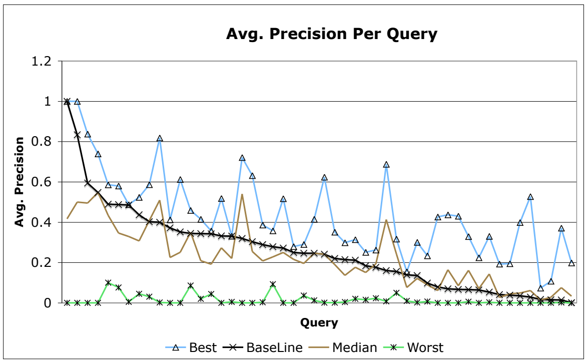

Figure 1: Avgerage Precision for each Query (ordered by relative performance) for the blog distillation task using a simple voting model. The best, median and worst scores are those of the other participants in TREC'08.

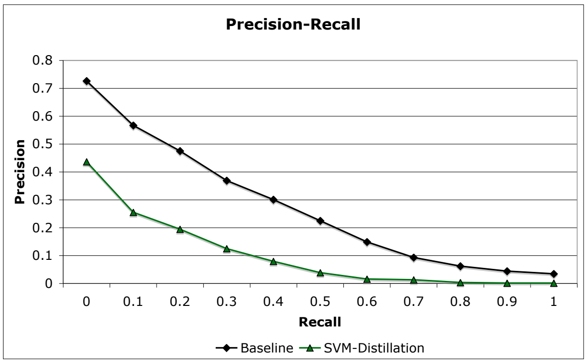

Figure 2: Precision-Recall for Baseline and Rank learning model

baselines. For each query we selected the 20,000 highest scoring postings as $R(Q)$ and use the voting model (Formula 5) to combine post relevance scores into blog relevance scores. Figure 1 shows the performance of our baseline for each query compared to best, worst and median precision for that query among all groups in TREC'08, where queries are sorted based on our baseline precision[4]. We saw that our baseline has reasonable performance across the queries and outperforms the median for most of the queries.

For the second experiment we extracted the features in Table 5 and used a rank learning system (SVM-map[5]) to learn a ranking function over them [18]. For the learning phase we trained the rank learner on the 45 topics and their corresponding relevance assessments used in the TREC'07 blog distillation task. We then tested the model on the 50 topics used for TREC'08.

Figure 2 shows the precision-recall curve for the trained ranking model compared to the baseline expert search approach. We see that using other features beside content doesn't help in retrieving better blogs. These results are consistent with results in [8], where the use of anchor text and cohesiveness didn't appear to improve performance. The reason is possibly due to the data set itself. Only 3% of postings have a link to another posting inside the data set and even in this small percent of links there are some noisy and non-informative links. Thus we will need to look for a better way to use this information. Furthermore the small amount of training data available - only 45 queries - may have affected the ability of the rank learner to learn a model that generalized well to the test data.

7. CONCLUSIONS AND FUTURE WORK

In the Opinion Retrieval task in TREC'08 we tried learning approach for generating opinion scores for individual documents and also for learning a ranking function that combines opinion evidence with relevancy into a single ranking function. A distinct advantage of our approach is that by performing multiple levels of learning, we maximize the use of available training data while not relying on external sources of opinion-bearing word lists, that may actually not be well-suited to the blog opinion retrieval task. In future we plan to extend the basic framework and consider the proximity of query terms to opinionated terms in assigning opinion weights to documents. We also plan to expand the feature space and capture more complicated opinion expressions by using bigrams or trigrams.

In the Distilation task we have implemented a baseline that has reasonable results compared with other systems. But we have learned that Blog Distillation performance does not seem to improve when using the information contained in links between posts in a blog data set. We conjecture that this is because some links are not useful and we should find a

[4] Jonathan Elsas suggested this representation in his weblog, `http://windowoffice.tumblr.com/`

[5] `http://projects.yisongyue.com/svmmap/`

way to use this link information only when probability of its informativeness is high. To do this, we plan to focus only on links to blogs that are related to the topic and analyse their structure. Since the most important blogs about a topic (those that contain the most information) will likely also be the most influential, other related blogs would link to them. So we can try to measure the influence of a blog in terms of the number of inlinks from blogs containing similar content. In this case similarity between the source and destination of a link will be taken into account in addition to the similarity between blogs and topics.

8. REFERENCES

[1] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. Automatic construction of an opinion-term vocabulary for ad hoc retrieval. In *Proceedings of the 30th European Conference on IR Research (ECIR)*, pages 89–100, 2008.

[2] G. Amati, C. Carpineto, and G. Romano. Fondazione ugo bordoni at trec 2004. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, 2004.

[3] G. Amati and C. J. V. Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS).*, 20(4):357–389, 2002.

[4] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, 2007.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[6] M. Efron, D. Turnbull, and C. Ovalle. University of Texas School of Information at TREC 2007. In *Proc. of the 2007 Text Retrieval Conf*, 2007.

[7] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and feedback models for blog feed search. In *SIGIR*, pages 347–354, 2008.

[8] 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*, 2007.

[9] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarks improve web search? In *WSDM '08: Proceedings of the First ACM International Conference on Web Search and Data Mining*, 2008.

[10] C. Macdonald and I. Ounis. The TREC Blogs06 collection: Creating and analysing a blog test collection. *Department of Computer Science, University of Glasgow Tech Report TR-2006-224*, 2006.

[11] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 387–396. ACM Press New York, NY, USA, 2006.

[12] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the trec-2007 blog track. In *Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[13] C. D. Manning and H. Schtze. *Foundations of Statistical Natural Language Processing*. The MIT Press, June 1999.

[14] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

[15] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49, 2004.

[16] J. Seo and W. Croft. UMass at TREC 2007 Blog Distillation Task. In *Proc. of the 2007 Text Retrieval Conf*, 2007.

[17] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can social bookmarking enhance search in the web? In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pages 107–116. ACM, 2007.

[18] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278, New York, NY, USA, 2007. ACM.

[19] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.