University of Glasgow at TREC 2009: Experiments with Terrier

Blog, Entity, Million Query, Relevance Feedback, and Web tracks

Richard McCreadie, Craig Macdonald, Iadh Ounis, Jie Peng, Rodrygo L.T. Santos

Department of Computing Science
University of Glasgow
Scotland, UK

{richardm,craigm,ounis,pj,rodrygo}@dcs.gla.ac.uk

ABSTRACT

In TREC 2009, we extend our Voting Model for the faceted blog distillation, top stories identification, and related entity finding tasks. Moreover, we experiment with our novel xQuAD framework for search result diversification. Besides fostering our research in multiple directions, by participating in such a wide portfolio of tracks, we further develop the indexing and retrieval capabilities of our Terrier Information Retrieval platform, to effectively and efficiently cope with a new generation of large-scale test collections.

1. INTRODUCTION

In TREC 2009, we participate in the Blog, Entity, Million Query, Relevance Feedback and Web tracks. This year, we have further developed our Terrier IR platform [29] with regards to efficiency and effectiveness for the newly introduced large-scale collections. Participation in such a wide portfolio of tracks allows us to comprehensively evaluate Terrier in a challenging environment. Our primary research directions focus on further applications for the Voting Model [20], as well as on experimenting with our novel xQuAD framework for search result diversification [34, 35].

In the faceted blog distillation task of the Blog track, we investigate how machine learning techniques can be used to address faceted blog ranking. In particular, on top of a Voting Model-based blog retrieval system, we devise a large set of features, and investigate the effectiveness of formulating the faceted blog distillation problem as a text classification or a learning-to-rank problem.

For the Blog track top news stories identification task, we identify the most important headlines for each day, by using the Voting Model. In particular, we believe that the number of blog posts mentioning a headline (aka votes) is a good indicator of the importance of each headline. However, as the blogosphere exhibits a bursty nature, we examine how to make use of the fact that important headlines can persist over a period of days. Lastly, we identify a set of novel yet relevant blog posts for each headline, by diversifying these blog posts based on temporal distance or content similarity.

In the Entity track, we extend the Voting Model to the task of finding related entities, by considering the co-occurrence of query terms and candidate entities in a document as a vote for the strength of the relationship between these entities and the query entity. In addition, we experiment with novel graph-based techniques, in order to promote entities associated to authoritative documents or documents from the same community as the query entity.

For the diversity task of the Web track, we experiment with our novel xQuAD diversification framework, based on the explicit account of the possible aspects underlying a query, in the form of sub-queries [34, 35]. In particular, we investigate the effectiveness of exploiting query suggestions provided by a major Web search engine as sub-queries within our proposed framework.

Lastly, in our participations in the Web track adhoc task, the Relevance Feedback track and the Million Query track, we test the effectiveness and efficiency of Terrier on the large-scale ClueWeb09 corpus. In particular, we test and further enhance our MapReduce-based indexing implementation in Terrier [27, 28], and deploy distributed retrieval techniques [6, 32] to permit efficient experimentation on this new, large-scale corpus.

The remainder of this paper is structured as follows. Section 2 describes the corpora used in our participation, along with the associated indexing and retrieval strategies we employ. Section 3 defines the models we use for retrieval and relevance feedback, and also introduces the Voting Model. Sections 4 and 5 cover our participation in the Blog track faceted blog distillation and top stories tasks, respectively. Our participation in the Entity track is discussed in Section 6. Sections 7 and 8 discuss our work in the adhoc and diversity tasks of the Web track, respectively. Sections 9 and 10 present our hypotheses and results for the Million Query and Relevance Feedback tracks, respectively. Lastly, Section 11 provides concluding remarks and directions for future research.

2. INDEXING & RETRIEVAL

The test collection for the Blog track is the new TREC Blogs08 collection, which is a crawl of the blogosphere over a 54-week period [25]. During this time, the blog posts (permalinks), feeds (RSS/Atom XML) and homepages of each blog were collected. In our participation in the Blog track, we index only the permalinks component of the collection. In particular, there are approximately 28 million documents in this component.

For the Entity, Million Query, Relevance Feedback, and Web tracks, the test collection is the new billion document TREC ClueWeb09 collection, which has an uncompressed size of 25TB.[1] We index this collection in two manners. Firstly, the so-called 'category B' subset, containing 50 million English documents, and secondly all 500 million English documents (the 'category A' subset).

[1] http://boston.lti.cs.cmu.edu/Data/clueweb09

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2009		**00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
University of Glasgow at TREC 2009: Experiments with Terrier	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Glasgow,Department of Computing Science,Scotland, UK,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009) held in Gaithersburg, Maryland, November 17-20, 2009. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA).

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**10**	

Stage	Input	Output
Map	Document	⟨Term, PostingList⟩
Reduce	Term, list[PostingList]	Inverted index

Table 1: Overview of the MapReduce functions used during indexing.

For indexing purposes, we treat the above two collections in the same way. Using the Terrier IR platform[2] [29], we create content-based indices, including the document body and title. Each term is stemmed using Porter's English stemmer, and standard English stopwords are removed. In both cases, we use our distributed MapReduce indexing implementation in Terrier [27, 28]. The indexing process is split into many 'map' tasks over the input data, followed by one or more reduce tasks to create the final inverted index shards. In particular, Table 1 gives an overview of the map and reduce functions used in our implementation. Each map task takes as input a document to be indexed, and processes that document, building up a miniature inverted index in memory. When memory is exhausted, the mini-inverted index is emitted from the map task to disk, in the form of ⟨Term, PostingList⟩ tuples. Each reduce task takes as input several posting lists for a given term, and merges these into the final inverted index. Note that the number of reduce tasks defines the number of inverted index shards created. For more details and comparative experiments, see [27].

We use a distributed version of Terrier to speed up retrieval for large corpora. In particular, we use distributed retrieval for retrieving documents from the ClueWeb09 category A corpus (500 million documents). Following [6, 32], our system uses one query server to serve results from one or more document-partitioned index shards, while a centralised query broker is responsible for passing the query to each query server, and merging the results. Moreover, the process for each query follows two phases. Firstly, the query is tokenised, and each term is passed to the query servers to obtain their local statistics for the term. These local statistics are merged by the broker, so that accurate global statistics are obtained. In the second phase, the query servers score and rank their documents, making use of the global statistics. Finally, the documents from each query server are merged into a single ranking by the broker. During merging, no score normalisation is necessary, as the retrieval approach applied by each query server is identical, using exactly the same global statistics.

3. MODELS

The main weighting model used in our TREC 2009 participation is the DPH model, which is derived from the Divergence From Randomness (DFR) framework [1]. Using DPH, the relevance score of a document *d* for a query *Q* is given by [2]:

$$score(d,Q) = \sum_{t \in Q} \frac{qtw(1-F)^2}{tf+1} \cdot \left(tf \cdot \log_2(tf \cdot \frac{avg\_l}{l} \frac{N}{TF})\right) + 0.5 \cdot \log_2(2\pi \cdot tf \cdot (1-F)) \quad (1)$$

where F is given by tf/l, tf is the within-document frequency, and l is the document length in tokens. avg_l is the average document length in the collection, N is the number of documents in the collection, and TF is the term frequency in the collection. Note that DPH is a parameter-free model, and therefore requires no particular tuning. qtw is the query term weight and is given by qtf/qtf_{max}, where qtf is the query term frequency and qtf_{max} is the maximum query term frequency among all query terms.

[2] http://www.terrier.org

3.1 Terms Dependence in the Divergence From Randomness Framework

Taking into account the dependence and proximity of query terms in documents can increase adhoc retrieval effectiveness. To this end, we use an extension of the DFR framework that can account for the dependence of query terms in documents [19, 30]. In general, when using a term dependence model, the score of a document d for a query Q is given as follows:

$$score(d,Q) = \sum_{t \in Q} qtw \cdot score(d,t) + \sum_{p \in Q_2} score(d,p) \quad (2)$$

where $score(d,t)$ is the score assigned to a query term t in the document d, p corresponds to a pair of query terms, and Q_2 is the set that contains all possible combinations of two query terms. In Equation (2), $\sum_{t \in Q} qtw \cdot score(d,t)$ can be estimated by any DFR weighting model, such as DPH. The $score(d,p)$ of a pair of query terms in a document is computed as follows:

$$score(d,p) = -\log_2(P_{p1}) \cdot (1 - P_{p2}) \quad (3)$$

where P_{p1} is the probability that there is a document in which a pair of query terms p occurs a given number of times. P_{p1} can be computed with any randomness model from the DFR framework, such as the Poisson approximation to the Binomial distribution. P_{p2} corresponds to the probability of seeing the query term pair once more, after having seen it a given number of times. P_{p2} can be computed using any of the after-effect models in the DFR framework. The difference between $score(d,p)$ and $score(d,t)$ is that the former depends on occurrences of the pair of query terms p, while the latter depends on occurrences of the query term t.

This year, for obvious efficiency reasons, we applied the pBiL randomness model [19], which does not consider the collection frequency of pairs of query terms. It is based on the binomial randomness model, and computes the score of a pair of query terms in a document as follows:

$$score(d,p) = \frac{1}{pf+1}\Big(-\log_2(l-1)! + \log_2 pf! + \log_2(l-1-pf)! - pf\log_2(p_p) - (l-1-pf)\log_2(p'_p)\Big) \quad (4)$$

where l is size of document d in tokens, $p_p = \frac{1}{l-1}$, $p'_p = 1 - p_p$, and pf is the frequency of the tuple p, i.e., the number of windows of size ws in document d in which the tuple p occurs.

3.2 Relevance Feedback

We use a term weighting model in the context of the Relevance Feedback (RF) track, and also for pseudo-relevance feedback (PRF) and collection enrichment (CE) [18, 21, 31] in the Blog track. The central idea behind PRF is to assume that the top documents returned for a query are relevant, while in RF, a few relevant documents are known. We can then learn from these feedback documents to improve retrieval performance through query expansion or term re-weighting. In particular, we apply the Bo1 term weighting model, derived from the DFR framework [1]. This model is based upon Bose-Einstein statistics and works in a similar fashion to Rocchio's relevance feedback method [33]. In Bo1, the informativeness $w(t)$ of a term is given by:

$$w(t) = tf_x \cdot \log_2 \frac{1+P_n}{P_n} + \log_2(1+P_n) \quad (5)$$

where tf_x is the frequency of the term t in the pseudo-relevant set, P_n is given by $\frac{TF}{N}$, TF is the frequency of t in the whole collection, and N is the number of documents in the collection.

3.3 Voting Model

The Voting Model [20] addresses the task of ranking document aggregates instead of individual documents. In TREC 2009, we consider different types of aggregates for specific tasks. In the faceted blog distillation task of the Blog track, blogs are represented by aggregates of blog posts, whereas in the top stories identification task, aggregates of blog posts are used to represent the days in which these blog posts are published. Lastly, in the Entity track, entities are represented by aggregates of the documents in which they occur.

In all cases, we consider the ranking of documents with respect to the query Q, which we denote $R(Q)$. The intuition is that a document aggregate ranking with respect to Q can be modelled as a voting process, using the retrieved documents in $R(Q)$. Specifically, every document in $R(Q)$ is possibly associated with one or more aggregates, and these associations act as votes for each aggregate to be relevant to Q. The votes for each aggregate are then appropriately combined to form the final ranking, taking into account the number of associated voting documents, as well as their relevance scores. Importantly, this model is extensible and general, and is not collection or topic dependent. It should be noted that, in practice, $R(Q)$ contains only a finite number of top documents, with the size of $R(Q)$ denoted $|R(Q)|$.

In [24], we defined twelve voting techniques for aggregating votes for candidate experts within the context of the expert search task, inspired by data fusion techniques and social choice theory. In this work, we use two voting techniques, namely Votes, and expCombMNZ. In Votes, the score of an aggregate C with respect to a query Q is given by:

$$score_{Votes}(C, Q) = |R(Q) \cap profile(C)| \quad (6)$$

where $|R(Q) \cap profile(C)|$ is the number of documents from the profile of the aggregate C that are in the ranking $R(Q)$.

The robust and effective expCombMNZ voting technique ranks aggregates by considering the sum of the exponential of the relevance scores of the documents associated with each aggregate. Moreover, it includes a component which takes into account the number of documents in $R(Q)$ associated to each aggregate, hence explicitly modelling the number of votes made by the documents for each aggregate. In expCombMNZ, aggregates are scored as:

$$score_{expCombMNZ}(C, Q) = |R(Q) \cap profile(C)| \cdot \sum_{d \in R(Q) \cap profile(C)} \exp(score(d, Q)) \quad (7)$$

where $score(d, Q)$ is the score of document d for query Q, as given by a standard weighting model, such as DPH (Equation (1)).

4. BLOG TRACK: FACETED BLOG DISTILLATION TASK

In the faceted blog distillation task, the goal is to produce a ranking of blogs for a given query that have a recurrent interest in the topic of the query, and that also fulfil a required facet. In particular, three facets are considered in this task: indepth, opinionated, and shallow [25]. For each query, participants are required to provide a baseline ranking, and two rankings fulfilling the possible inclinations of the facet associated to the query. For instance, for an indepth-related query, besides a baseline ranking, participants should produce a second ranking, aimed at favouring indepth blogs, and a third ranking, aimed at favouring shallow blogs.

In TREC 2009, we deploy different machine learning techniques in order to identify blogs fulfilling a desired facet inclination, from a baseline ranking produced by the Voting Model. In particular, we investigate both traditional text classification techniques [36] as well as a state-of-the-art learning-to-rank technique [39] in order to produce targeted rankings for each inclination.

Our first approach to this task builds upon traditional text classification. By using four different classifiers, we estimate the extent to which a given blog matches the facet inclination of interest. In particular, we use the following classifiers: Naive Bayes, a decision tree learner (J48), logistic regression, and a Support Vector Machine (SVM) classifier [10]. The classifier's confidence in the classification of a blog to a particular inclination is then integrated with the baseline relevance score using FLOE [9]. In our second approach, we employ the AdaRank [39] learning-to-rank algorithm to produce a ranking model for each inclination.

To enable both approaches, we devise a set of 18 features, calculated from individual blog posts as well as entire blogs, for the facets considered in this task. For example, intuitively, long posts or sentences should reflect a more indepth blog, whereas having only a single author or having offensive words should likely constitute positive indicators of a personal blog. Additionally, for the opinionated facet, we repurpose our effective post-level opinion detection techniques [12, 13], deployed in previous Blog tracks [11, 14], in order to produce blog-level opinion features. Despite being motivated by our intuitions regarding specific facet inclinations, we do not restrict the use of these features to the identification of blogs fulfilling these inclinations, but instead let our deployed approaches decide whether and how to use each feature. Additionally, for our learning-to-rank approach, negated versions of all features are also considered, so as to allow the learner to decide whether a highly weighted feature should be considered a positive or a negative indicator of a particular facet inclination (for instance, is a long average sentence length a good or bad feature for an indepth blog?). Finally, both text classification and learning-to-rank approaches are trained using a few annotated examples of blogs that fulfil each facet inclination, gathered from the TREC Blogs06 collection [22].

We submit four runs to the faceted blog distillation task, as described next and summarised in Table 2. All runs use the DPH weighting model (Equation (1)) and the expCombMNZ voting technique (Equation (7)) to create an initial ranking of blogs, which are then re-ranked to match a particular facet inclination.

1. **uogTrFBNclas** uses the confidence scores provided by a naive Bayes classifier to re-rank blogs for each facet inclination.
2. **uogTrFBMclas** is similar to uogTrFBNclas, except that it uses the scores provided by the best (rather than a single) of our considered classifiers on a per-facet inclination basis, according to their performance on the training data.
3. **uogTrFBAlr** uses the AdaRank algorithm to learn a different ranking model for each facet inclination.
4. **uogTrFBHlr** is similar to uogTrFBAlr, but uses intuitively set feature weights for each facet inclination, as a baseline.

Table 3 shows the results of our submitted runs for each of the facets of interest. Performance is given in terms of mean average precision (MAP) on a per-facet inclination basis. Additionally, the performance of our baseline ranking for each inclination is also shown. Unfortunately, we had an oversight on the configuration of this baseline, which used an extremely large $|R(Q)|$, markedly

	Indepth (18 queries)				Opinionated (13 queries)				Personal (8 queries)			
	Base	First	Base	Second	Base	First	Base	Second	Base	First	Base	Second
TREC best	–	0.3489	–	0.1906	–	0.2338	–	0.2945	–	0.2995	–	0.3167
TREC median	–	0.0549	–	0.0250	–	0.0727	–	0.0685	–	0.0937	–	0.0560
uogTrFBNclas												
submitted	0.1033	0.0652	0.0256	0.0259	0.1012	0.0988	0.0954	0.0925	0.0981	0.0691	0.1691	0.1693
corrected	0.1671	0.0846	**0.0479**	0.0321	0.1074	0.1032	0.1266	0.1134	**0.1938**	0.1219	**0.1733**	0.1710
overfitted	0.1671	**0.1735**	**0.0479**	0.0113	0.1074	**0.1274**	0.1266	**0.1312**	**0.1938**	0.1813	**0.1733**	0.1636
uogTrFBMclas												
submitted	0.1033	0.0652	0.0256	0.0259	0.1012	0.0988	0.0954	0.0925	0.0981	0.0691	0.1691	0.1693
corrected	0.1671	0.1671	**0.0479**	0.0321	0.1074	0.1032	0.1266	0.0942	**0.1938**	0.1219	**0.1733**	0.1143
overfitted	0.1671	**0.2032**	**0.0479**	0.0113	0.1074	**0.1274**	0.1266	**0.1420**	**0.1938**	0.1813	**0.1733**	0.0903
uogTrFBAlr												
submitted	0.1033	0.0005	0.0256	0.0001	0.1012	0.0796	0.0954	0.1095	0.0981	0.0642	0.1691	**0.1743**
corrected	**0.1671**	0.0059	**0.0479**	0.0255	**0.1074**	0.0422	0.1266	0.1254	**0.1938**	0.0947	0.1733	0.0848
overfitted	**0.1671**	0.1589	**0.0479**	0.0444	**0.1074**	0.1048	0.1266	**0.1271**	**0.1938**	0.1227	0.1733	0.1521
uogTrFBHlr												
submitted	0.1033	0.1015	0.0256	0.0301	0.1012	0.0919	0.0954	0.1103	0.0981	0.0739	0.1691	**0.1965**
corrected	**0.1671**	0.1565	**0.0479**	0.0468	**0.1074**	0.0793	0.1266	**0.1276**	**0.1938**	0.1172	0.1733	0.1805
overfitted	–	–	–	–	–	–	–	–	–	–	–	–

Table 3: Per-facet MAP performance: submitted, corrected, and overfitted runs.

Run	Description
uogTrFBNclas	DPH+expCombMNZ+Naive
uogTrFBMclas	DPH+expCombMNZ+BestClass
uogTrFBAlr	DPH+expCombMNZ+AdaRank
uogTrFBHlr	DPH+expCombMNZ+Human

Table 2: Submitted runs to the faceted blog distillation task of the Blog track.

compromising the performance of our submitted runs. Hence, in Table 3, besides the performance of each run, with $|R(Q)|$ = 20,000, we include an additional row showing its attained performance after correcting the baseline ranking, with $|R(Q)|$ = 1,000. Additionally, in order to assess the impact of the used training data, Table 3 also includes a row with the performance of our runs when overfitted using the relevance assessments for this task.[3]

From Table 3, we first observe that the performance of our *submitted* runs is above median across most settings. Moreover, when the *corrected* runs are considered, improvements in terms of baseline performance are observed across all settings. The inclination-specific performance of these runs, in turn, increases across most settings, with the second inclination of the personal facet being the only exception. Nevertheless, even after correcting our baseline, re-ranking it in order to favour blogs fulfilling a desired facet inclination remains challenging. We hypothesise that this is partially due to the insufficient training data we had available. Indeed, when the *overfitted* runs are considered, a more comparable performance to that of our baseline ranking is observed for most settings. As for the deployed approaches themselves, the classification-based runs performed generally better for the first inclination of the indepth and personal facets, as well as for both inclinations of the opinionated facet, whereas our approach based on learning-to-rank was generally the best for the remaining settings. Overall, our results attest the difficulty of the task [25], but they also show some promising directions for improvement. In particular, the availability of suitable training data should allow us to better estimate the usefulness of different features in discriminating between blogs fulfilling different facet inclinations.

[3]Note that this training regime is not applicable for the uogTrFBHlr run, as it is independent of the used training data.

5. BLOG TRACK: TOP STORIES IDENTIFICATION TASK

In the top stories identification task, the goal is to produce a set of important headlines (from an editorial perspective) and associated blog posts in relation to a day of interest. In particular, the task involves, for each query day, finding the most important headlines for that day, and then selecting ten relevant and diverse blog posts for each of those headlines [25]. We divide the problem into two distinct sub-tasks: *headline ranking*, the ranking of top headlines for the query day; and *blog post selection*, where we select a diverse set of top blog posts pertaining to a headline.

For our participation in this task, we investigate the application of the Voting Model [20] (see Section 3.3) to the headline ranking problem. For blog post selection for a given headline, we explore diversity by promoting relevant yet novel blog posts in the ranking. In particular, we explore both the textual and temporal dissimilarity between blog posts as evidence for diversification.

5.1 Headline Ranking

The aim of the headline ranking sub-task is to produce a set of headlines which were deemed, from an editorial perspective, to be important on the query day d_Q, using evidence from the blogosphere. Our headline ranking approach is based on the intuition that, on any day, bloggers will create posts pertaining to prominent news stories for that day. We desire to score a headline h for a given query *day* d_Q, which we denote $score(h, d_Q)$. Our basic approach uses the Votes voting technique (Equation (6)) to score all headlines published on day $d_Q \pm 1$ (to account for the time difference between countries), by counting the number of blog posts mentioning the headline h on query day d_Q (i.e. from the ranking of blog posts $R(h)$). We use DPH (Equation (1)) for ranking blog posts in response to a headline. As suggested in [20], we limit the number of retrieved blog posts to $|R(h)| \leq 1000$.

However, blog posts created after the query day d_Q may also help to improve the accuracy of our approach. Our intuition is that news stories will often be discussed afterwards for long running, controversial or important unpredictable stories, e.g. the aftermath of a terrorist bombing. Indeed, by taking this evidence into account, we can identify those stories which maintain their interest over time, and as such can be deemed more important. In particular, [16] suggested that bursts in term distributions could last for a period of time. Hence, in the following, we define two alternative

Run	Headline Ranking	Blog Post Selection
uogTrTsbmmr	DPH + Votes. $\|R(h)\|$=1000 (baseline)	DPH + Diversify(MMR)
uogTrTswtime	+ NDayBoost(n=7)	DPH + Diversify(Time)
uogTrTstimes	+ CE(Wikipedia,10 terms)	MergedSubRankings(DPH) + Diversify(Time)
uogTrTSemmrs	+ CE(Wikipedia,10 terms) + GaussBoost(w=4,n=14)	MergedSubRankings(DPH) + Diversify(MMR)

Table 4: Summary of submitted runs to the top stories identification task of the Blog track.

techniques for calculating $score(h, d_Q)$, which leverage the *temporal distribution* of each headline h over time. In particular, these techniques accumulate vote evidence from the days preceding or following d_Q, to 'boost' the score of headlines which retain their importance over multiple days.

In our first proposed temporal distribution boosting technique, $NDayBoost$, we linearly combine the scores for the following n days before or after day d_Q, as:

$$score_{NDayBoost}(h, d_Q) = \sum_{d=d_Q}^{d_Q+n} |R(h, d)| \quad (8)$$

where $|R(h, d)|$ measures the importance of headline h on day d, n is a parameter controlling the number of days before ($n < 0$) or after ($n > 0$) d_Q to take into account, while d represents any single day. Note that this technique places equal emphasis on all days d – we expect the distribution of $|R(h, d)|$ to peak around day d_Q.

Importantly, this approach can incorporate evidence from multiple days. However, due to the linear nature of the score aggregation, all days are treated equally, when it is intuitive to think that days more distant from d_Q will provide poorer evidence.

To address this, we propose a second temporal distribution boosting technique. In particular, $GaussBoost$ is similarly based upon the intuition that important stories will run for multiple days. However, instead of judging each subsequent day equally, we weight based on the time elapsed from the day of interest d_Q, using a Gaussian curve to define the magnitude of emphasis. In this way, we state a preference for stories that were most important around d_Q, rather than stories which peaked some time before/after d_Q:

$$score_{GaussBoost}(h, d_Q) = \sum_{d=d_Q}^{d_Q+m} Gauss(d - d_Q) \cdot |R(h, d)| \quad (9)$$

where m is the maximum number of days before or after d_Q to take into account and $d - d_Q$ is the number of days elapsed since the day of interest d_Q ($0 \leq d_Q \leq m$). $Gauss(\Delta d)$ is the Gaussian curve value for a difference of days Δd, as given by:

$$Gauss(\Delta d) = \frac{1}{w.\sqrt{2\pi}} \cdot \exp \frac{-(\Delta d)^2}{(2w)^2} \quad (10)$$

where w defines the width of the Gaussian curve. A smaller w will emphasise stories closer to d_Q, while a larger w will take into account stories on more distant days, up to the maximum m days.

It should also be noted that the original headlines provided for this task contain many non-news entries (e.g. paid death notices, corrections, etc). We apply a small set of heuristics to the headline corpus beforehand to remove these spurious entries, on the intuition that these headlines can never be deemed important. Furthermore, as a means to counter term sparsity in the headlines, we investigate the usefulness of collection enrichment [18, 21, 31] in this domain. Indeed, expanding queries based on a higher quality, external resource has been shown to be more effective than doing so on the local collection, since blog posts are often noisy [14]. In particular, we enrich each headline from Wikipedia (as extracted from the ClueWeb09 collection) using DPH (Equation (1)) for retrieval and Bo1 (Equation (5)) to select the top 10 terms for each headline. Our submitted runs are summarised in Table 4.

Run	Submitted	Corrected
TREC best	0.2600	
TREC median	0.0400	
uogTrTsbmmr	0.1731	N/A
uogTrTswtime	0.0795	0.1812
uogTrTstimes	**0.1862**	N/A
uogTrTSemmrs	0.1186	0.1720

Table 5: Headline ranking MAP performance of our submitted and corrected (where applicable) runs for the top stories identification task of the Blog track.

Table 5 presents the mean average precision for headline ranking over our four submitted runs. From the results, we see that our baseline (uogTrTsbmmr) voting-based approach provides a strong performance of 0.1731 MAP, which is markedly higher than the median for this task. Indeed, all of our submitted runs comfortably exceed this median. Note that, for our boosting runs (uogTrTswtime and uogTrTSemmrs), we encountered a 'long' to 'int' overflow bug, which affected their performance. Once this was corrected, their performances were comparable to our baseline, as shown in the corrected column of Table 5. Indeed, uogTrTswtime improved upon our baseline ranking, indicating that there is useful evidence which can be leveraged to improve the ranking performance from after the query day. Our best run was that done with collection enrichment using Wikipedia, which indicates that, indeed, term sparsity within headlines is an important factor, and deserves further investigation. Moreover, uogTrTswtime proved to be the best run at the TREC 2009 Blog top stories task [25].

5.2 Blog Post Selection

The goal of the blog selection sub-task is to retrieve a set of ten blog posts for a given headline which are both relevant to this headline, and moreover cover as large a variety of the aspects of this headline as possible. Using DPH, we produce a first ranking of blog posts for each headline. However, there is also additional temporal information which can be exploited to improve upon this initial ranking. During the headline selection sub-task, our approach generates day-oriented blog post rankings for each headline – i.e. for day d_Q, the top blog posts (if any) which match each retrieved headline h. We exploit this to create a second, enhanced blog post ranking, by merging some of these day-oriented blog post rankings together, keeping only the top scored results. In particular, we merge the rankings for the day of the headline, with those for the following week. In this way, we restrict the blog posts to be selected to only those in temporal proximity to the query day d_Q, on the intuition that these will more likely be relevant, while still bringing potentially novel information as the story develops.

To diversify either of these two blog post rankings, we then apply one of two re-ranking techniques: diversification through textual dissimilarity and diversification using temporal dissimilarity. For textual dissimilarity, we apply the Maximal Marginal Relevance (MMR) [7] method. In particular, MMR greedily selects a docu-

Run	α-NDCG@10	IA-P@10
TREC best	0.7723	0.2758
TREC median	0.0217	0.0040
uogTrTsbmmr	**0.518**	**0.168**
uogTrTswtime	0.297	0.094
uogTrTstimes	0.449	0.155
uogTrTSemmrs	0.371	0.123

Table 6: Blog post selection performance of our submitted runs for the top stories identification task of the Blog track.

ment d^* from the initial ranking with maximum relevance to the query (headline) and maximum dissimilarity to the previously selected documents (blog posts). The selection criterion used by the MMR algorithm is defined below:

$$d^* = \arg\max_{d_i \in R \setminus S} [\lambda \, \mathrm{Sim}_1(d_i, h) - (1 - \lambda) \max_{d_j \in S} \mathrm{Sim}_2(d_i, d_j)] \tag{11}$$

where R is a ranked list of blog posts, h is a headline, S is the subset of documents in R already selected, and $R \setminus S$ is the set difference, i.e, the documents not yet selected. Sim_1 is the similarity metric used in document retrieval (i.e. DPH), and Sim_2 is the similarity between documents d_i and d_j, which can be computed by the same metric used for Sim_1 or a different one. In particular, we use the cosine distance between vector representations of the blog posts d_i and d_j, weighted by DPH.

For temporal dissimilarity, we develop a novel time-based diversification approach, which exploits the evolution of a story over time. The intuition is that, as the story progresses, different viewpoints will be expressed and new actors will arrive. Hence, to truly provide an overview of a particular story, we hypothesise that blog posts should be selected over time. To promote a wide variety of blog posts over the course of the story, we select blog posts with increasing temporal distance from the headline time. In particular, we incrementally select blog posts published at least 6 hours apart. Our submitted runs are listed in the last column of Table 4.

Our results are shown in Table 6. We can see that all our results outperform the TREC median by a large margin, with our best run (uogTrTsbmmr) achieving 0.518 α-NDCG@10. Indeed, it was the best top news stories identification task run at TREC 2009 [25]. Moreover, both maximal marginal relevance (uogTrTsbmmr) and temporal diversification (uogTrTstimes) proved to be effective techniques when applied on our baseline DPH blog post ranking. In contrast, runs using our merged blog post rankings (uogTrTswtime and uogTrTSemmrs) were less effective. However, it is unclear whether their performance is due to the method itself, or to the input data from the headline ranking, which was the subject of the overflow bug mentioned earlier. In point of fact, further investigation confirmed that indeed the input data was to blame.

6. ENTITY TRACK

In the new Entity track, the goal is to retrieve entities of a particular type (people, organisations, or products) that are somehow related to an input entity in the query [4]. Our major goal in this track was to extend our Voting Model to the task of finding related entities of the desired type. Our approach builds a semantic relationship support for the Voting Model, by considering the co-occurrences of query terms and entities within a document as a vote for the relationship between these entities and the one in the query. Additionally, on top of the Voting Model, we develop novel techniques to further enhance the initial vote estimations. In particular, we promote entities associated to authoritative documents or documents from the same community as the query entity in the hyperlink structure underlying the ClueWeb09 collection.

Firstly, in order to identify entities in the category B subset of the corpus, we resort to an efficient dictionary-based named entity recognition approach.[4] In particular, we build a large dictionary of entity names using DBPedia,[5] a structured representation of Wikipedia. Dictionary entries comprise all known aliases for each unique entity, as obtained from DBPedia (e.g., 'Barack Obama' is represented by the dictionary entries 'Barack Obama' and '44th President of the United States'). In order to differentiate between the entity types of interest in this task, DBPedia names are further categorised as people, organisations, or products, based on each entity's category description on DBPedia and several heuristics (for instance, the occurrence of the clue word 'company' is likely to identify organisations). In order to account for people that do not have a Wikipedia page, entries in the produced dictionary are complemented with common proper names derived from the US Census data.[6] After being identified, entity name occurrences in the corpus are recorded in appropriate index structures, so as to make this information efficiently available at querying time. By doing so, a rich profile is built for every unique entity, comprising the documents in which the entity occurs in the corpus.

Additionally, in order to find the correct homepages for each retrieved entity, we again resort to DBPedia. In particular, for some catalogued entities, DBPedia includes a set of associated documents, which correspond to external (i.e., non-Wikipedia) pages linked to from each entity's Wikipedia page, and are more likely to correspond to the desired homepages for that entity. For entities with no such associated documents and also for non-DBPedia entities, we simply retrieve the top scored documents from the entities' profile as their candidate homepages.

At querying time, we experiment with different approaches that refine the initial ranking of documents for a given query. Firstly, on top of the DPH weighting model (Equation (1)), we apply the pBiL proximity model (Equation (4)), in order to favour documents in which the query terms occur in close proximity. This can be particularly beneficial, as the queries in this task include named entities. Additionally, in an attempt to promote authoritative homepages at the document ranking level, we integrate a document indegree feature, computed on the hyperlink graph underlying the category B subset of the ClueWeb09 collection. Alternatively, we experiment with a state-of-the-art community detection technique [5], in order to favour documents from the same community as those associated to the input entity. By doing so, we expect to promote the entities associated to these documents, with the intuition that these entities are more likely to be related to the input entity. On top of the document ranking produced by either of these techniques, the expCombMNZ voting technique (Equation (7)) is then applied to produce a ranking of entities, generating the following runs:

1. **uogTrEbl** is a baseline run, which applies the DPH weighting model and the expCombMNZ voting technique.
2. **uogTrEpr** applies the pBiL proximity model at the document ranking level, with a window size $ws = 4$.
3. **uogTrEdi** integrates the indegree feature to the document ranking using FLOE, with the settings suggested in [9].
4. **uogTrEc3** promotes entities associated to documents from the same community as those associated to the input entity.

[4] `http://alias-i.com/lingpipe`
[5] `http://dbpedia.org`
[6] `http://www.census.gov/genealogy/names/names_files.html`

Table 7 summarises our submitted runs, while Table 8 presents their results in terms of normalised discounted cumulative gain (NDCG) at R, where R is the number of primary and relevant documents (i.e., homepages), precision at 10 (P@10), and the total number of relevant (rel) and primary (pri) homepages retrieved.

Run	Description
uogTrEbl	DPH+expCombMNZ
uogTrEpr	DPH+pBiL+expCombMNZ
uogTrEdi	DPH+indegree+expCombMNZ
uogTrEc3	DPH+communities+expCombMNZ

Table 7: Submitted runs to the Entity track.

From the results in Table 8, we observe that all our runs perform well above the median of the participant groups according to both NDCG@R and P@10. Moreover, all four runs achieved by far the best performance among the TREC participants in terms of the number of relevant and primary homepages associated with the retrieved entities [4], hence attesting the strength of our baseline approach. Additionally, the integration of the document indegree feature further improved over our strongly performing baseline in terms of P@10. Moreover, applying proximity at the document ranking level brought improvements in terms of both NDCG@R and P@10, as did our community-based boosting technique. Overall, these results not only attest the effectiveness of our Voting Model extension for this task, but also demonstrate its promise as a general framework for entity-related search tasks.

Run	NDCG@R	P@10	#rel	#pri
TREC best	0.4098	0.3500		
TREC median	0.0751	0.0050		
uogTrEbl	0.2510	0.1050	344	75
uogTrEpr	0.2662	0.1200	347	79
uogTrEdi	0.2502	0.1150	343	74
uogTrEc3	0.2604	0.1200	331	75

Table 8: Results of submitted runs to the Entity track.

7. WEB TRACK: ADHOC TASK

In the adhoc task of the Web track, participants aimed to identify topically relevant documents on both the category B (50 million documents) and category A (500 million documents) subsets of the ClueWeb09 corpus [8]. In this task, we aimed to test our DFR models, and the Terrier IR platform on this larger corpus.

In particular, we submitted three runs to the adhoc task. Two of these were for category B, one for category A. For all runs, we applied the DPH DFR model (Equation (1)). In particular, the submitted runs, and an unsubmitted baseline are described below:

- **uogTrdph** is our unsubmitted baseline, and uses DPH only.
- **uogTrdphP** adds the pBiL proximity model (Equation (4)), with window size $ws = 4$, to the scores generated by DPH.
- **uogTrdphA** tests the simple use of anchor text, by uniformly combining scores from content and anchor text indices.
- **uogTrdphCEwP** uses collection enrichment (CE) [18, 21, 31], by expanding the queries from documents retrieved only from the Wikipedia portion of ClueWeb09. The Bo1 term weighting model (Equation (5)) is used to weight terms in the pseudo-feedback documents. Additionally, the pBiL proximity model is also applied by this run, with $ws = 4$.

A summary of our submitted runs is given in Table 9. Their retrieval performance is provided in Table 10. For the category B runs, we note the following: our DPH weighting model baseline run (uogTrdph) performed well above median; applying collection enrichment and proximity (uogTrdphCEwP) improved upon the baseline; however, the simplistic combination of anchor text with content used by run uogTrdphA was detrimental to retrieval performance. For the category A runs, our retrieval performance was roughly median. On a closer inspection of our category A run, we found that it suffered from retrieving many spam Web pages. Hence, in the future, we will investigate the application of techniques to remove spam, and/or identify high quality documents.

Run	Submitted	Cat.	Description
uogTrdph	✘	A/B	DPH(content)
uogTrdphP	✔	A	+pBiL
uogTrdphA	✔	B	+DPH(anchor text)
uogTrdphCEwP	✔	B	+CE+pBiL

Table 9: Submitted and unsubmitted runs to the adhoc task of the Web track, including the category of each run.

B runs	Submitted	statMAP	statNDCG
TREC best		0.4305	0.6091
TREC median		0.1539	0.2956
uogTrdph	✘	0.1970	0.3096
uogTrdphA	✔	0.1825	0.3245
uogTrdphCEwP	✔	0.2072	0.3934
A runs		P@5	P@10
TREC best		0.8320	0.7780
TREC median		0.1600	0.1720
uogTrdph	✘	0.0650	0.0969
uogTrdphP	✔	0.1600	0.1660

Table 10: Results of submitted and unsubmitted runs to the adhoc task of the Web track.

Overall, our results in this task show the promise of the Terrier IR platform and the DFR weighting models for larger corpora, even without any training, since our participation in TREC 2009 relied solely on parameter-free models. Additionally, we believe we can enhance our retrieval performance by applying field-based weighting models (e.g. PL2F [19]), and, particularly on the A subset, developing spam detection techniques.

8. WEB TRACK: DIVERSITY TASK

The goal of the diversity task of the Web track is to produce a ranking of documents that (1) maximises the coverage and (2) reduces the redundancy of the retrieved documents with respect to the possible aspects underlying a query, in the hope that users will find at least one of these documents to be relevant to this query [8].

In our participation in this task, we propose to explicitly take into account the possible aspects underlying a query, in the form of sub-queries [34, 35]. By estimating the relevance of the retrieved documents to individual sub-queries, we seek to produce a re-ranking of these documents that maximises the coverage of the aspects underlying the initial query, while reducing its redundancy with respect to already well covered aspects. In particular, we experiment with our novel framework for search result diversification, called **xQuAD** (eXplicit Query Aspect Diversification). Given a query Q, and an input ranking $R(Q)$, xQuAD iteratively builds a result ranking $S(Q)$ by selecting, at each iteration, the document $d^* \in R(Q) \setminus S(Q)$ with the highest score, as given by:

$$d^* = \underset{d \in R(Q) \setminus S(Q)}{\arg\max}\ r_1(d,Q) \sum_{Q' \in G(Q)} \frac{i(Q',Q) r_2(d,Q')}{m(Q', S(Q))} \quad (12)$$

where:

- $r_1(d,Q)$ is the relevance of document d with respect to the initial query Q, as estimated by any retrieval approach, such as the DPH document weighting model (Equation (1)),
- $G(Q)$ is the set of sub-queries Q' associated to Q,
- $i(Q',Q)$ is the estimated importance of the sub-query Q' relatively to all sub-queries associated to Q,
- $r_2(d,Q')$ is the relevance of document d to the sub-query Q', as estimated by any retrieval approach (not necessarily the same used for $r_1(d,Q')$), and
- $m(Q', S(Q))$ estimates the amount of information satisfying the sub-query Q' present in the documents already selected in $S(Q)$, as a measure of novelty.

In our experiments, the $G(Q)$ component is based on query suggestions provided by a major Web search engine for each of the TREC 2009 Web track topics. Alternatively, we investigate a new cluster-based query expansion technique aimed at generating sub-queries from the target collection itself. In particular, we cluster the top retrieved results for an initial query using the k-means algorithm [26], and then generate different sub-queries by expanding the initial query from each individual cluster.

As for the importance component, $i(Q',Q)$, we propose a simple baseline estimation mechanism, $i_u(Q',Q)$, which considers a uniform importance distribution over sub-queries:

$$i_u(Q',Q) = \frac{1}{|G(Q)|}, \quad (13)$$

where $|G(Q)|$ is the number of sub-queries generated for query Q. Alternatively, we experiment with biasing the diversification process towards those sub-queries likely to represent more plausible aspects of the initial query. Inspired by a state-of-the-art resource selection technique [37], we estimate the relative importance of each generated sub-query, by considering the ranking produced for this sub-query as a sample of the documents it covers in the whole collection. In particular, we estimate the importance $i_c(Q',Q)$ of the sub-query Q' as:

$$i_c(Q',Q) = \frac{n(Q')}{\max_{Q'_i \in G(Q)} n(Q'_i)} \frac{1}{\hat{n}(Q')} \sum_{d | r_2(d,Q') > 0} \tau - j(d,Q), \quad (14)$$

where $r_2(d,Q')$ is as described above, $n(Q')$ is the total number of results associated with the sub-query Q', $\hat{n}(Q')$ corresponds to the number of results associated to Q' that are among the top τ ranked results for the initial query Q, with $j(d,Q)$ giving the ranking position of the document d with respect to Q.

Finally, the novelty component $m(Q', S(Q))$ is estimated as the number of documents retrieved for the sub-query Q' that are among the already selected documents in $S(Q)$.

In our submitted runs, we use the DPH weighting model to produce the initial baseline ranking, and also the ranking for each identified sub-query, for category A. For category B, DPH is used along with the pBiL proximity model, with a window size $ws = 4$. On top of the initial baseline, we experiment with the different components of our proposed framework to produce diverse rankings with $\tau = 1000$ documents, resulting in the following three submitted runs, summarised in Table 11:

Run	Cat.	$r_{\{1,2\}}(d,Q)$	$G(Q)$	$i(Q',Q)$
uogTrDYScdA	A	DPH	sWQ	i_u
uogTrDPCQcdB	B	DPH+pBiL	cQE	i_u
uogTrDYCcsB	B	DPH+pBiL	sWQ	i_c

Table 11: Submitted runs to the diversity task of the Web track, including the category of each run.

1. **uogTrDYScdA** retrieves documents from the whole of ClueWeb09, and then re-ranks these using query suggestions from a major Web search engine as sub-queries (denoted sWQ), weighted by the i_u importance estimator.
2. **uogTrDPCQcdB** investigates generating sub-queries from the collection itself, by applying the previously described cluster-based query expansion technique (denoted cQE). The Bo1 term weighting model (Equation (5)) is used to produce sub-queries from each cluster generated by k-means ($k = 10$) from a baseline ranking of 1000 documents. In our experiments, a maximum of 10 terms are expanded from the 3 highly scored documents in each cluster, so as to form a sub-query. The i_u importance estimator is used once again.
3. **uogTrDYCcsB** uses the same sub-queries as uogTrDYScdA (i.e., sWQ), but with the importance of each sub-query estimated by our resource selection-inspired technique, i_c.

Table 12 shows the performance of our runs in the diversity task, in terms of α normalised discounted cumulative gain (α-NDCG) and intent-aware precision (IA-P). From the table, we observe that all our runs perform well above the median of the TREC participants, for both category A and B settings, and in terms of both measures. Indeed, run uogTrDYCcsB was the best performing among all participating runs for category B, in terms of both α-NDCG and IA-P [8]. Notwithstanding, there is still scope for improvements, as demonstrated by a further analysis of the individual components underlying our framework, their own performance, and their contribution to the performance of the approach as a whole [35].

Run	Cat.	α-NDCG@10	IA-P@10
TREC best	A	0.5144	0.2105
TREC median	A	0.1324	0.0541
uogTrDYScdA	A	0.1910	0.0770
TREC best	B	0.5267	0.2447
TREC median	B	0.1758	0.0733
uogTrDPCQcdB	B	0.2710	0.1340
uogTrDYCcsB	B	0.2820	0.1320

Table 12: Retrieval performances of our submitted runs to the diversity task of the Web track.

9. MILLION QUERY TRACK

In the Million Query track, participants aimed to identify topically relevant documents for many queries, on both the category B (50 million documents) and category A (500 million documents) subsets of the ClueWeb09 corpus. We submitted two runs to the Million query track, to see how the performance of these runs differed from the performance of the equivalent runs submitted to the adhoc task of the Web track. The runs were:

- **uogTRMQdph40** applies DPH on the content-only index (40,000 queries). This run corresponds to run uogTrdph from the adhoc task of the Web track (see Section 7).

- **uogTRMQdpA10** applies DPH, combining scores from body and anchor text indices (10,000 queries). This run corresponds to run uogTrdphA from the adhoc task of the Web track (see Section 7).

Table 13 summarises the obtained retrieval performance of the runs submitted to the Million Query track. We note that the retrieval performance of both runs are markedly above the median measures, particularly on the statMAP measure. Anchor text makes no marked benefit to eMAP performance, but is detrimental to statMAP performance, mirroring our observations from Section 7.

Run	statMAP	eMAP
TREC best	0.5378	0.1592
TREC median	0.1535	0.0591
uogTRMQdpA10	0.2612	0.0869
uogTRMQdph40	0.2339	0.0881

Table 13: Summary of retrieval performance of our submitted Million Query track runs.

10. RELEVANCE FEEDBACK TRACK

The aim of the TREC 2009 Relevance Feedback track was to examine the aspects affecting the selection of good feedback documents. In our participation in this track, we focus on the stability of our query expansion Bo1 DFR term weighting model (Equation (5)) across different feedback identification strategies.

In the first phase of the track, participants submitted 5 ClueWeb09 category B documents to be assessed for each topic. Our first feedback set, ugTr.1, was created using the DLH13 model [21]. Our second feedback set, ugTr.2, was created using the DPH model (and hence corresponds to run uogTrdph from Section 7). Table 14 reports the P@5 performances of our submitted feedback sets, and also of several other phase 2 feedback sets. From the table, we note that the ugTr.1 compared well with the other feedback sets, while ugTr.2 was the best performing of this selection of feedback sets.

Feedback Set	P@5
ugTr.1	0.320
ugTr.2	0.504
CMU.1	0.340
UMas.1	0.496
UPD.1	0.460
YUIR	0.252
hit2.1	0.320
ilps.2	0.368

Table 14: Retrieval performances of our submitted phase 1 feedback sets, and our phase 2 allocated feedback sets for the Relevance Feedback track.

In the second phase, participants submitted one run for each of the 8 feedback sets assigned to them. We used only the relevant documents from each feedback set, and ranked documents from the category B collection using the DPH document weighting model (i.e. based on the Web track adhoc task baseline uogTrdph). We used the Bo1 term weighting model (Equation (5)) to identify and weight the 10 most informative terms to expand the query with.

Unfortunately, our second phase relevance feedback runs encountered a bug in our mapping from 'DOCNO' to internal docid, and as a consequence, the actual used feedback documents were incorrect. Hence, in the following, we report the performance of our submitted runs, and the correct retrieval performances. Table 15 reports the performance of these submitted and corrected runs,[7] for the various feedback sets. In all cases, retrieval performance was detrimentally impacted by the presence of the bug, compared to the corrected runs. Moreover, we note that Bo1 does not appear to favour the feedback sets identified by the DFR weighting models (i.e. ugTr.1 & ugTr.2). Indeed, some of the less accurate feedback sets (see Table 14) perform better overall. This suggests that some of these sets produce better feedback documents for Bo1 than DPH or DLH13 alone [15]. Indeed, Bo1 performed best using the hit2.1 feedback set, even though hit2.1 performed worse than ugTr.2 (hit2.1 exhibited 37% less P@5 than ugTr.2).

Run	Submitted		Corrected
	statMAP	eMAP	statMAP
ugTr.CMU.1	0.1764	0.0409	0.2492
ugTr.UMas.1	0.1958	0.0421	0.2373
ugTr.UPD.1	0.1715	0.0399	0.2325
ugTr.YUIR	0.1900	0.0460	0.2055
ugTr.hit2.1	0.1667	0.0414	0.2578
ugTr.ilps.2	0.1756	0.0407	0.2212
ugTr.ugTr.1	0.2081	0.0464	0.2240
ugTr.ugTr.2	0.1810	0.0409	0.2349

Table 15: Retrieval performances of our allocated feedback sets for the Relevance Feedback track.

Overall, we conclude that our participation in the Relevance Feedback track has facilitated an investigation into the aspects affecting the Bo1 term weighting model, and testified to its suitability for application using various methods for generating feedback sets.

11. CONCLUSIONS

In TREC 2009, we participated in five tracks, namely the Blog, Entity, Million Query, Relevance Feedback and Web tracks, using our Terrier IR platform. In particular, our participation focused on new applications for the Voting Model, as well as on fresh approaches for search result diversification.

Our results for the Blog track top news stories identification task are particularly strong. In the faceted blog distillation task, a configuration oversight hindered the retrieval performance of our runs. Nevertheless, our corrected results show very good promise, but also attest that this task remains hard without suitable training data. In the Entity track, our proposed extension to the Voting Model has been shown to provide a very effective framework for tackling the related entity finding task. In the diversity task of the Web track, the new xQuAD framework shows a strong retrieval performance, with promising directions for further improvements.

Finally, with the advent of several larger test collections, we took the opportunity presented by TREC 2009 to overhaul the Terrier platform, for instance with improved MapReduce indexing, and scalable retrieval. However, a small bug in the improved system affected our submitted Relevance Feedback track runs. On the larger ClueWeb09 category A collection, our runs were affected by the presence of spam in the collection. In the future, we will endeavour to develop spam detection and document quality features.

Acknowledgements

We would like to thank Ben He for his help with the Relevance Feedback runs, and Stuart Chalmers for assisting us in our TREC assessment workload this year. We are also grateful to Duncan McDougall for helping us with the extraction and preparation of several features used in the Blog and Entity tracks.

[7] eMAP measures cannot be calculated by participants.

12. REFERENCES

[1] G. Amati. *Probabilistic Models for Information Retrieval based on Divergence From Randomness*. PhD thesis, Dept. of Computing Science, Univ. of Glasgow, 2003.

[2] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog track. In *Proceedings of TREC 2007*.

[3] G. Amati, C. Carpineto, and G. Romano. Italian monolingual information retrieval with Prosit. In *Proceedings of CLEF 2001*.

[4] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 Entity Track. In *Proceedings of TREC 2009*.

[5] V.D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of community hierarchies in large networks. In *Proceedings of CoRR 2008*.

[6] F. Cacheda, V. Plachouras, and I. Ounis. A case study of distributed information retrieval architectures to index one terabyte of text. *IP&M*, 41(5), 2005.

[7] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR 1998*.

[8] C. L. A. Clarke, N. Craswell, and I. Soboroff. Preliminary report on the TREC 2009 Web track. In *Proceedings of TREC 2009*.

[9] N. Craswell, S. Robertson, H. Zaragoza, and M. Taylor. Relevance weighting for query independent evidence. In *Proceedings of SIGIR 2005*.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations, 11(1), 2009*.

[11] 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*.

[12] B. He, C. Macdonald, and I. Ounis. Ranking opinionated blog posts using OpinionFinder. In *Proceedings of SIGIR 2008*.

[13] B. He, C. Macdonald, J. He, and I. Ounis. An effective statistical approach to blog post opinion retrieval. In *Proceedings of CIKM 2008*.

[14] B. He, C. Macdonald, I. Ounis, J. Peng, and R. L. T. Santos. University of Glasgow at TREC 2008: experiments in Blog, Enterprise, and Relevance Feedback tracks with Terrier. In *Proceedings of TREC 2008*.

[15] B. He and I. Ounis. Finding good feedback documents. In *Proceedings of CIKM 2009*.

[16] J. Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of KDD 2002*.

[17] K. L. Kwok, L. Grunfeld, M. Chan, N. Dinstl, and C. Cool. TREC-7 Ad-Hoc, High Precision and Filtering experiments using PIRCS. In *Proceedings of TREC 1998*.

[18] K. L. Kwok and M. Chan. Improving two-stage ad-hoc retrieval for short queries. In *Proceedings of SIGIR 1998*.

[19] C. Lioma, C. Macdonald, V. Plachouras, J. Peng, B. He, and I. Ounis. University of Glasgow at TREC 2006: experiments in Terabyte and Enterprise tracks with Terrier. In *Proceedings of TREC 2006*.

[20] C. Macdonald. *The Voting Model for People Search*. PhD thesis, Dept. of Computing Science, Univ. of Glasgow, 2009.

[21] C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of Glasgow at TREC 2005: experiments in Terabyte and Enterprise tracks with Terrier. In *Proceedings of TREC 2005*.

[22] C. Macdonald and I. Ounis. The TREC Blogs06 collection: creating and analysing a blog test collection. Tech. report TR-2006-224, Dept. of Computing Science, Univ. of Glasgow, 2006.

[23] C. Macdonald and I. Ounis. Key blog distillation: ranking aggregates. In *Proceedings of CIKM 2008*.

[24] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of CIKM 2006*.

[25] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2009 Blog track. In *Proceedings of TREC 2009*.

[26] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of Berkeley SMSP 1967*.

[27] R. McCreadie, C. Macdonald, and I. Ounis. Comparing distributed indexing: to MapReduce or not? In *Proceedings of LSDS-IR Workshop at SIGIR 2009*.

[28] R. McCreadie, C. Macdonald, and I. Ounis. On single-pass indexing with MapReduce. In *Proceedings of SIGIR 2009*.

[29] 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 OSIR Workshop at SIGIR 2006*.

[30] J. Peng, C. Macdonald, B. He, V. Plachouras, and I. Ounis. Incorporating term dependency in the DFR framework. In *Proceedings of SIGIR 2007*.

[31] J. Peng, B. He, and I. Ounis. Predicting the Usefulness of Collection Enrichment for Enterprise Search. In *Proceedings of ICTIR 2009*.

[32] V. Plachouras, B. He, and I. Ounis. University of Glasgow at TREC 2004: experiments in Web, Robust and Terabyte tracks with Terrier. In *Proceedings of the TREC 2004*.

[33] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System, 1971*.

[34] R. L. T. Santos, J. Peng, C. Macdonald, and I. Ounis. Explicit search result diversification through sub-queries. In *Proceedings of ECIR 2010*.

[35] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for Web search result diversification. In *Proceedings of WWW 2010*.

[36] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv., 34(1), 2002.*.

[37] M. Shokouhi. Central-Rank-based Collection Selection in uncooperative distributed information retrieval. In *Proceedings of ECIR 2007*.

[38] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. OpinionFinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005*.

[39] J. Xu and H. Li. AdaRank: a boosting algorithm for information retrieval. In *Proceedings of SIGIR 2007*.