# University of Glasgow at TREC 2010: Experiments with Terrier in Blog and Web tracks

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

School of Computing Science
University of Glasgow
G12 8QQ, Glasgow, UK

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

## ABSTRACT

In TREC 2010, we continue to build upon the Voting Model and experiment with our novel xQuAD framework within the auspices of the Terrier IR Platform. In particular, our focus is the development of novel applications for data-driven learning in the Blog and Web tracks, with experimentation spanning hundreds of features. In the Blog track, we propose novel feature sets for the ranking of blogs, news stories and blog posts. In the Web track, we propose novel selective approaches for adhoc and diversity search.

## 1. INTRODUCTION

In TREC 2010, we participate in the Blog track faceted blog distillation and top story identification tasks, as well as the Web track adhoc and diversity tasks. Our focus is the development of novel applications for data-driven learning to each of these tasks using the Terrier IR platform [11], increasing effectiveness through large-scale experiments using hundreds of individual features.

In the blog distillation task of the Blog track, we deploy machine learning techniques to learn both the ranking of blogs for a query, and their inclination given the facet of the query. For the top news stories identification task, we build upon our effective voting approach and experiment with data-driven learning both to rank news stories for a single day, and also to rank blog posts for a single story.

The major goal of our participation in the Web track is to investigate novel data-driven selective approaches, based on a large set of document and query features. In the adhoc task, we seek to determine, on a per-query basis, the most appropriate ranking model to be applied. In the diversity task, for each query, we determine not only whether to diversify, but also by how much.

## 2. BLOG TRACK: FACETED BLOG DISTILLATION TASK

We investigate novel data-driven approaches for both the baseline and faceted blog distillation tasks of the Blog track. In particular, in the baseline blog distillation task, the aim is to identify blogs which have a principle, recurring interest in the query topic, while in the faceted task, these blogs should be further ranked with respect to a facet inclination of interest, namely opinionated, factual, indepth, shallow, personal, and official.

### 2.1 Baseline Blog Distillation

In our participation to the baseline blog distillation task, we extend the Voting Model [5], which has previously been shown to be effective for identifying key blogs [6]. In particular, the Voting Model specifies many *voting techniques*, each of which aggregates evidence from a single ranking of blog posts to produce a ranking of blogs. However, instead of using a single blog post ranking and a single voting technique, we propose a novel approach to learn the aggregation of different rankings of blog posts with multiple voting techniques [7]. As a result, we mix the qualities of different voting techniques into a learned ensemble [7]. A total of 450 voting technique features are combined using the Metzler's Automatic Feature Selection (AFS) learning to rank technique [10], trained on the TREC 2009 blog distillation task.

We submitted two runs to the baseline blog distillation task, summarised below:

- uogTrapeMN5k: expCombMNZ voting technique, using the top 5000 blog posts ranked by DPH.
- uogTrLv450: learned ranking, combining many voting techniques, using a total of 450 features.

| Run | MAP | P@10 |
|---|---|---|
| TREC median | 0.1925 | 0.3097 |
| uogTrapeMN5k | 0.2024 | 0.2009 |
| uogTrLv450 | 0.2001 | 0.2055 |

**Table 1: Results of the submitted runs to the baseline blog distillation task.**

The results of our submitted runs are given in Table 1. From the results, we note that both of our submitted runs outperformed the TREC median MAP. Moreover, we find that our learned approach to blog distillation is promising, as uogTrLv450 successfully improves over the early precision of the uogTrapeMN5k run.

### 2.2 Faceted Blog Distillation

Following our data-driven theme, for the faceted blog distillation task we also apply machine learning techniques for identifying and appropriately ranking the facet inclination of a blog.

In particular, to identify the facet inclination of every retrieved blog, we deploy many features, including blog post-level and blog-level features. For instance, the number of inlinks, or the presence of opinionated terms [4] are examples of blog post-level features that we deploy. Additionally, inspired by the work of He et al. [4] at identifying opinionated terms, we identify a dictionary of terms for each facet inclination using the relevance assessments of the TREC 2009 faceted blog distillation task. Using all of the terms in

the dictionaries for each facet inclination, facet inclination feature scores for each blog post are obtained. All features are combined with the baseline blog retrieval score, in two different manners: In our first approach, we combine all features with the baseline blog retrieval score using a learning to rank approach. Secondly, we use a classifier to build a classification model using all features for each facet inclination, before integrating the confidence of the classifier with the baseline retrieval score.

We submitted four groups of runs to the faceted blog distillation task, using four baselines runs (uogTrfL728, stdbaseline1, stdbaseline2 - which is uogTrapeMN5k from Section 2.1 - and stdbaseline3). Our groups of runs, which are summarised below, use both learning to rank and classification approaches to facet ranking, and mix feature sets with and without the use of dictionary features:

- uogTrfL728: learned ranker, based on 728 features.
- uogTrfL919: learned ranker, based on 728 features, plus an additional 191 dictionary features for facet inclinations, totalling 919 features in all.
- uogTrfC728: Using the same 728 features as uogTrfL728, but using a classifier to permit the re-ranking of results by their classified inclination.
- uogTrfC919: As uogTrfC728, but using the same feature set as uogTrfL919.

Table 2 details the performance of our submitted faceted blog distillation runs, in terms of MAP for each facet inclination, and the mean over all inclinations. From the results, we make the following observations and conclusions:

- Our learned approaches, namely uogTrfL728 & uogTrfL919 generally perform higher than the classification approaches (uogTrfC728 & uogTrfC919).
- Comparing the number of features (728 vs. 919), we note that the used feature set has a different impact according to the deployed learning techniques and baseline.
- Runs based on the first of the TREC provided baselines, namely stdbaseline1, perform the best for each group of runs. This highlights the importance of a strong baseline, as this is the highest performing of the baselines that we deploy.

Overall, we conclude that our feature sets and learned approaches are effective for faceted blog distillation.

# 3. BLOG TRACK: TOP STORIES IDENTIFICATION TASK

In the top stories identification task, the goal is to produce a set of important stories for a day in question, as well as a high quality and diversified ranking of blog posts for those stories. In particular, the task comes in two distinct stages, namely *news story ranking* and *blog post ranking*. During news story ranking, for a set of query days, the stories published on each day are to be ranked by their newsworthiness on that day for each of five news categories, namely U.S., World, Sport, Business and Science/Technology. This tasks mimics a real-time setting, where blog post evidence after the time of the query cannot be used. For blog post ranking, given a set of news stories, blog posts are ranked based upon their relevance for these stories, as well as in terms of their diversity in covering different aspects of each story.

## 3.1 News Story Ranking

In the top stories ranking task, we adopt a data-driven learning approach. In particular, we learn how to rank stories by their predicted importance based on the blogosphere, by inferring the magnitude of blogging activities as well as the usefulness of story representations as features. In particular, we assume that bloggers will create posts pertaining to prominent news stories for each day. Therefore, we consider that the relative magnitude of this posting activity in comparison to previous days is indicative of a story's importance on those days. To measure this blogging activity, we employ two effective voting techniques - firstly, ranking stories by their votes from blog posts [8] (referred to as *Votes*), and secondly a new voting technique, referred to as *Relevance Weighted Aggregation* (RWA), which accounts for both the relevance of blog posts in addition to their volume.

To classify each news story into the task categories, we leverage crowdsourcing to create training labels for an open source ngram language model classifier provided by LingPipe[1]. In particular, we use Amazon's Mechanical Turk[2] to label 3000 randomly sampled news stories from days predating the Blogs08 timespan. We integrate the classification labels in the category ranking in three different regimes: strict, lax and balanced. Strict considers stories to belong to only the most likely category, lax classifies stories into multiple likely categories using a low threshold upon the classifier confidence for each category, while balanced similarly classifies each story into multiple classes using a higher threshold.

We submitted 3 story ranking task runs. In particular, we submitted one baseline run (uogTrCh) which uses only our best voting technique, and two learned runs using either 151 restricted (safer) features or 1076 features respectively. These learned runs were trained on the 2009 top news stories ranking topics using Metzler's Automatic Feature Selection (AFS) learning to rank technique [10]. Note that each run uses a different classification regime. Our submitted runs are as follows:

- uogTrCh: Our Relevance Weighted Aggregation voting technique, upon the headline alone using a balanced classifier.
- uogTrLC151: A learned run using an intuitive set of 151 features from RWA. These features represent two story representations (headline and content), story ranking evidence from the two days preceding the query day and varying time ranges from which voting blog posts can be selected. Stories were classified by the strict classifier.
- uogTrLV1076: A learned run, using 1076 features produced from Votes and RWA. These features encompass eight different story representations, story ranking evidence from the seven days preceding the query day and varying time ranges from which voting blog posts can be selected. Stories were classified using the lax classifier.

Table 3 reports story ranking performance of our submitted runs in comparison to the TREC best systems. In particular, column 2 reports the story ranking performance under the official relevance assessments, i.e. after stories have been classified into the five news categories, while column 3 reports story ranking performance in general, i.e. pre-classification. To calculate pre-classification performance, we assume that if a story was important to any category then it was important overall. From Table 3, we observe that the post-classification (official) performance of our submitted runs is lower than anticipated. However, we also see from column 3, that

[1] `http://alias-i.com/lingpipe`
[2] `http://www.mturk.com`

| Run | Baseline | Mean Facet MAP | MAP by Facet | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | opinionated | factual | official | personal | indepth | shallow |
| uogTrfC728 | uogTrLv450 | 0.0873 | 0.0883 | 0.0493 | 0.0461 | 0.0729 | 0.1261 | 0.1409 |
| uogTrfC728s1 | stdbaseline1 | 0.1383 | 0.2539 | 0.0783 | 0.1006 | 0.0619 | 0.2298 | 0.1051 |
| uogTrfC728s2 | stdbaseline2 | 0.1045 | 0.0841 | 0.2002 | 0.0729 | 0.0373 | 0.1070 | 0.1255 |
| uogTrfC728s3 | stdbaseline3 | 0.0619 | 0.0641 | 0.0395 | 0.0673 | 0.0415 | 0.0760 | 0.0828 |
| uogTrfC919 | uogTrLv450 | 0.0890 | 0.1044 | 0.0426 | 0.0570 | 0.0707 | 0.1100 | 0.1496 |
| uogTrfC919s1 | stdbaseline1 | 0.1386 | 0.2406 | 0.0797 | 0.0983 | 0.0747 | 0.2347 | 0.1037 |
| uogTrfC919s2 | stdbaseline2 | 0.0958 | 0.1053 | 0.1296 | 0.0739 | 0.0365 | 0.1312 | 0.0981 |
| uogTrfC919s3 | stdbaseline3 | 0.0711 | 0.1001 | 0.0341 | 0.0602 | 0.0549 | 0.0854 | 0.0921 |
| uogTrfL728 | uogTrLv450 | 0.1058 | 0.0885 | 0.1661 | 0.0907 | 0.0876 | 0.1256 | 0.0761 |
| uogTrfL728s1 | stdbaseline1 | 0.1730 | 0.2417 | 0.1365 | 0.1486 | 0.1012 | 0.2971 | 0.1129 |
| uogTrfL728s2 | stdbaseline2 | 0.1026 | 0.0991 | 0.1847 | 0.0728 | 0.0499 | 0.1258 | 0.0835 |
| uogTrfL728s3 | stdbaseline3 | 0.0815 | 0.0487 | 0.1323 | 0.0541 | 0.0713 | 0.0913 | 0.0916 |
| uogTrfL919 | uogTrLv450 | 0.0982 | 0.0875 | 0.0540 | 0.1140 | 0.0954 | 0.1309 | 0.1076 |
| uogTrfL919s1 | stdbaseline1 | 0.1837 | 0.2440 | 0.1369 | 0.2456 | 0.1017 | 0.2578 | 0.1162 |
| uogTrfL919s2 | stdbaseline2 | 0.1067 | 0.0800 | 0.1804 | 0.1105 | 0.0546 | 0.1333 | 0.0811 |
| uogTrfL919s3 | stdbaseline3 | 0.0769 | 0.0477 | 0.0973 | 0.0843 | 0.0524 | 0.0980 | 0.0819 |

**Table 2: Results of the submitted runs to the faceted blog distillation task.**

| Run | Post-Classification (official) statMAP | Pre-Classification statMAP |
|---|---|---|
| TREC median | 0.1361 | 0.1355 |
| TREC 1st | 0.2206 | 0.1898 |
| TREC 2nd | 0.2151 | 0.1730 |
| TREC 3rd | 0.2138 | 0.1497 |
| uogTrCh | 0.0828 | 0.1866 |
| uogTrLC151 | 0.0360 | 0.1812 |
| uogTrLV1076 | 0.0466 | 0.1759 |

**Table 3: Pre-classification and post-classification story ranking performance in terms of statMAP.**

in terms of pre-classification story ranking, our runs offer similar performance to that attained by the TREC best systems. This indicates that while our unlearned model is effective at ranking news stories, our classifier requires further improvement.

Furthermore, we observe that pre-classification, our learned models (uogTrLC151 and uogTrLV1076) are less effective than the baseline. Analysis of these runs indicate that this results from poor feature generalisation between the 2009 and 2010 topics. For example, the most effective story representation (the strongest ranking feature) on the 2009 topics was the headline, while on the 2010 topics, the content was markedly more effective. Indeed, when ranking with the content, the unlearned model can achieve a pre-classification performance of 0.2080 statMAP, higher than the best TREC systems.

## 3.2 Blog Post Ranking

To rank blog posts with regard to a news story, we similarly employ a data-driven approach, this time to learn the features of blog posts that are most useful when ranking with regard to story relevance. In this way, we aim to improve upon our effective DPH baseline blog post ranking used during TREC 2009. In particular, we leverage 81 different blog post features. Our proposed approach learns the extent to which each of these features is useful when ranking blog posts for a news story. In addition, we leverage information from the set of entities covered by the story, as well as the possible facet inclinations of the blog posts, in order to produce a diverse ranking within our xQuAD framework [18].

We submitted 3 blog post ranking task runs. In particular, we submitted one adhoc learned run, and two diversified runs, each using a different representation for the aspects of each story.

- uogTrL81: A learned model which uses 81 blog post features, including retrieval, link-based and opinionatedness metrics. This model was trained on the TREC 2009 blog post ranking topics using AFS [10].
- uogTrdxE: A DPH ranking explicitly diversified using xQuAD, with different story aspects represented by extracted entities.
- uogTrdxF: As uogTrdxE, except that story aspects are represented by different facet inclinations.

| Run | $\alpha$-nDCG@10 |
|---|---|
| TREC median | 0.421 |
| uogTrL81 | 0.477 |
| uogTrdxF | 0.413 |
| uogTrdxE | 0.404 |

**Table 4: $\alpha$-nDCG@10 performance for our blog post ranking runs for the TREC Blog Track top stories identification task.**

Table 4 reports the performance of our three submitted runs in terms of the official TREC measure ($\alpha$-nDCG@10). We observe that our learned model that uses 81 blog post features (uogTrL81) is effective at ranking blog posts related to each news story. Indeed, uogTrL81 markedly outperforms the TREC median for this task. On the other hand, our diversified runs do not perform as effectively as the learned model, indicating that neither our identified entities nor facet inclinations are sufficient to represent the aspects underlying a news story and its related blog posts.

# 4. WEB TRACK: ADHOC TASK

In the adhoc task, we use a novel framework for selective information retrieval. Our novel selective framework automatically decides which ranking model (from a set of candidate learned models) is the most appropriate for an unseen query [13]. Similarly to [17], we use many query features to decide on the best ranking model, given an unseen query.

In our participation, we use learning to rank to obtain effective candidate models and runs. In particular, we create learned models using pools of 42 and 67 features, summarised in Table 5. All models are learned using TREC 2009 Web track training data and the AFS learning to rank technique [10]. Our novel selective framework chooses the most appropriate model for each query, based on over 700 query features, also summarised in Table 5.

Three runs were submitted to the adhoc task:

- uogTrA42 (cat. A) deploys ranking models learned on Web queries using document features selected from pools of 42 features.

| Groups | | Document Features | Total | Query Features (most from [17]) | Total |
|---|---|---|---|---|---|
| 42 feat. | 67 feat. | Weighting models (DPH [1], PL2 [1], BM25 [15]) | 25 | NGram features | 11 |
| | | Fields-based models (BM25F [19], PL2F [1]) | 2 | Query ambiguity | 121 |
| | | URL and link analysis features (e.g. PageRank, Absorbing Model [14]) | 14 | Query log mining | 14 |
| | | Spam feature (Cormack's fusion score [2]) | 1 | Query performance predictors | 7 |
| | | Term-dependence models (MRF [9], pBiL [12]) | 25 | Taxonomy-based features | 604 |

**Table 5: Document and query features used in the Web track.**

| Run | Cat. | P@5 | P@10 | nDCG@20 | ERR@20 |
|---|---|---|---|---|---|
| TREC median | | - | - | 0.1412 | 0.0805 |
| uogTrA42 | A | 0.3875 | 0.4104 | 0.2446 | 0.1267 |
| uogTrB67 | B | 0.4250 | 0.4062 | 0.2097 | 0.1191 |
| uogTrB67* | B | 0.4208 | 0.4021 | 0.2572 | 0.1413 |
| uogTrB67LTS | B | 0.4042 | 0.4083 | 0.1899 | 0.1136 |

**Table 6: Results of the submitted runs to the adhoc task of the Web track. Corrected run is denoted with *.**

- uogTrB67 (cat. B) deploys ranking models learned on Web queries using document features selected from pools of 67 features.
- uogTrB67LTS (cat. B) deploys our novel selective framework for automatically selecting an appropriate ranking model on a per-query basis.

Table 6 shows the results of our submitted runs to the adhoc task. From the table, we observe that all runs perform markedly above the TREC median. However, we later found that our anchor text and URL representations were not indexed correctly. To address this issue, Table 6 also shows the performance of an equivalent corrected run to uogTrB67, denoted uogTrB67*. We can see that the corrected run uogTrB67* improves nDCG@20 and ERR@20 compared to uogTrB67. uogTrB67LTS, which deploys an appropriate ranking model on a per-query basis, attains the highest cat. B P@10, attesting the effectiveness of our novel selective framework.

## 5. WEB TRACK: DIVERSITY TASK

Our participation in the diversity task builds upon our state-of-the-art xQuAD framework [16, 17, 18]. Based on an initial ranking $R$ for the query $q$, xQuAD iteratively builds a re-ranking $S$ by selecting, at each iteration, a document $d^* \in R \setminus S$ such that:

$$d^* = \arg\max_{d \in R \setminus S} (1-\lambda)\Pr(d|q) + \lambda \Pr(d, \bar{S}|q), \quad (1)$$

where $\Pr(d|q)$ is the probability of a document $d$ satisfying the query $q$ and $\Pr(d, \bar{S}|q)$ is the probability of this document but none of the documents already in $S$ satisfying $q$. In practice, these two probabilities can be thought of as representing the *relevance* and the *diversity* of $d$, respectively, with the parameter $\lambda$ controlling the trade-off between the two probabilities. Additionally, the probability $\Pr(d, \bar{S}|q)$ can be further expanded according to:

$$\Pr(d, \bar{S}|q) = \sum_{s_i \in Q} \Pr(s_i|q) \Pr(d|q, s_i) \prod_{d_j \in S} \Pr(\bar{d_j}|q, s_i), \quad (2)$$

where the sub-query $s_i \in Q$ represents one of the multiple possible aspects underlying the query $q$, $\Pr(s_i|q)$ represents the importance of this sub-query in light of $q$, $\Pr(d|q, s_i)$ estimates the coverage of $d$ with respect to $s_i$, and $\prod \Pr(\bar{d_j}|q, s_i)$ estimates the novelty of any document satisfying $s_i$, in terms of how badly this sub-query is satisfied by the previously selected documents $d_j \in S$.

In TREC 2010, we have two main research directions. Firstly, we investigate whether xQuAD can be improved by enhancing the estimations of $\Pr(d|q)$ (i.e., the relevance component) and $\Pr(d|q, s_i)$ (i.e., the coverage and novelty components) using learning-to-rank (LTR). Secondly, we investigate whether setting the trade-off parameter $\lambda$ on a per-query basis can further improve xQuAD's performance. The latter research direction entails a selective approach to search result diversification, whereby we decide, for each query, not only whether to diversify, but also by how much [17].

For our submitted runs, we generate sub-queries for each of the TREC 2010 queries based on query reformulations from Bing and Google [18]. For learning the relevance, coverage, and novelty components, we leverage the same document features used for our adhoc runs described in Section 4. For predicting the diversification trade-off $\lambda$, we deploy two different regimes:

- UNI, where we uniformly set the same $\lambda$ value for all queries, based on the optimal value observed using all the 50 TREC 2009 queries for training.
- SEL, where we selectively set $\lambda$ for each individual query $q$ as the average optimal $\lambda$ value observed for the three most similar queries to $q$ from TREC 2009.

For the SEL regime, similar training queries are identified using a $k$NN classifier and the 757 query features described in Table 5.

We produced a total of ten runs in the diversity task, three of which were officially submitted as per our participation:

- uogTrA42 (A, submitted) is a LTR adhoc run, as described in Section 4.
- uogTrA42x (A, submitted) applies xQuAD using uogTrA42 as the relevance component, with coverage and novelty estimated by DPH, and the trade-off $\lambda$ set uniformly.
- uogTrBdph (B, unofficial) is an adhoc run based on DPH.
- uogTrBdphx (B, unofficial) applies xQuAD using uogTrBdph as the relevance component, with coverage and novelty estimated by DPH, and the trade-off $\lambda$ set uniformly.
- uogTrBdphxS (B, submitted) is similar to uogTrBdphx, except that the trade-off $\lambda$ is set selectively.
- uogTrB67 (B, submitted) is a learning-to-rank adhoc run, as described in Section 4.
- uogTrB67x (B, unofficial) applies xQuAD using uogTrB67 as the relevance component, with coverage and novelty estimated by DPH, and the trade-off $\lambda$ set uniformly.
- uogTrB67xS (B, submitted) is similar to uogTrB67x, except that the trade-off $\lambda$ is set selectively.
- uogTrB67lx (B, unofficial) is similar to uogTrB67x, except that the diversity components are based on LTR.
- uogTrB67lxS (B, unofficial) is similar to uogTrB67lx, except that the trade-off $\lambda$ is set selectively.

| Run | Cat. | Rel. | Div. | $\lambda$ | ERR-IA @20 | $\alpha$-nDCG @20 | NRBP @1000 | Submitted? |
|---|---|---|---|---|---|---|---|---|
| TREC median | | | | | 0.1947 | 0.3117 | – | |
| uogTrA42 | A | LTR | – | – | 0.2220 | 0.3214 | 0.1860 | adhoc |
| uogTrA42x | A | LTR | DPH | UNI | 0.2454 | 0.3558 | 0.2012 | diversity |
| uogTrBdph | B | DPH | – | – | 0.1774 | 0.2833 | 0.1295 | *unofficial* |
| uogTrBdphx | B | DPH | DPH | UNI | 0.2428 | 0.3574 | 0.2005 | *unofficial* |
| uogTrBdphxS | B | DPH | DPH | SEL | 0.2830 | 0.4051 | 0.2393 | diversity |
| uogTrB67 | B | LTR | – | – | 0.2981 | 0.4177 | 0.2616 | adhoc |
| uogTrB67x | B | LTR | DPH | UNI | 0.3142 | 0.4319 | 0.2758 | *unofficial* |
| ~~uogTrB67xS~~ | ~~B~~ | ~~LTR~~ | ~~DPH~~ | ~~SEL~~ | ~~0.2981~~ | ~~0.4178~~ | ~~0.2616~~ | ~~diversity~~ |
| uogTrB67xS | B | LTR | DPH | SEL | 0.3056 | 0.4357 | 0.2637 | *unofficial* |
| uogTrB67lx | B | LTR | LTR | UNI | 0.3098 | 0.4374 | 0.2680 | *unofficial* |
| uogTrB67lxS | B | LTR | LTR | SEL | 0.3184 | 0.4440 | 0.2784 | *unofficial* |

**Table 7: Results of the submitted runs to the diversity task of the Web track.**

Table 7 shows the results of our unofficial as well as our officially submitted runs to the diversity task. The struck out line indicates a bug in the submitted uogTrB67xS run, which mistakenly used the wrong predicted $\lambda$ values. The table is organised into three main groups, according to the run that served as the adhoc baseline in each case (i.e., uogTrA42, uogTrdph, and uogTrB67).

In the first group, we observe that xQuAD (uogTrA42x) successfully improves upon the adhoc baseline (uogTrA42) according to all considered measures. In the second group, we note that xQuAD (uogTrdphx) also improves upon the adhoc baseline (uogTrBdph), with further improvements observed when the selective regime is deployed (uogTrBdphxS). In the third group, similar results are observed when deploying xQuAD uniformly (uogTrB67x) or selectively (uogTrB67xS) on top of the adhoc baseline (uogTrB67). Analysing the impact of learning the coverage and novelty components, we observe that an estimation of these components based on LTR (uogTrB67lx) only improves compared to when DPH is used (uogTrB67x) in terms of $\alpha$-nDCG@20, with slight decreases in terms of the other measures. However, when the selective regime is considered, using LTR brings further improvements (uogTrB67lxS vs. uogTrB67xS), with uogTrB67lxS attaining our overall best performance. In fact, based on the preliminary evaluations of all participants' runs (i.e., with 36 of the final 48 topics), uogTrB67lxS (ERR-IA@20 = 0.367, $\alpha$-nDCG@20 = 0.509) would have ranked just above the top-performing run, uwgym (ERR-IA@20 = 0.356, $\alpha$-nDCG@20 = 0.500), which was produced by querying a commercial search engine [3]. This observation further attests the effectiveness of our xQuAD framework [16, 18] and the potential of enhancing its underlying components, whether through learning-to-rank or our proposed selective diversification approach [17].

## 6. CONCLUSIONS

In TREC 2010, we participated in the Blog and Web tracks using our Terrier IR platform. In particular, our participation focused around novel data-driven approaches, as well as improved applications of the Voting Model, enhanced search result diversification using xQuAD, and new selective approaches to ranking Web documents. Our results attest the effectiveness of our deployed machine learning approaches to both Blog and Web retrieval tasks.

## 7. REFERENCES


[1] 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*.

[2] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. In *Information Retrieval*, 2011.

[3] C. L. A. Clarke, N. Craswell, I. Soboroff, and G. V. Cormack. Preliminary overview of the TREC 2010 Web track. In *Proceedings of TREC 2010*.

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

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

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

[7] C. Macdonald and I. Ounis. Learning Models for Ranking Aggregates. In *Proceedings of ECIR 2011*.

[8] R. McCreadie, C. Macdonald, and I. Ounis. News Article Ranking: Leveraging the Wisdom of Bloggers. In *Proceedings of RIAO 2010*.

[9] D. Metzler. A Markov random field model for term dependencies. In Proceedings of SIGIR 2005.

[10] D. Metzler. Automatic feature selection in the Markov random field model for information retrieval. In *Proceedings of CIKM 2007*.

[11] 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*.

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

[13] J. Peng, C. Macdonald, and I. Ounis. Learning to Select a Ranking Function. In *Proceedings of ECIR 2010*.

[14] V. Plachouras, I. Ounis, and G. Amati. The Static Absorbing Model for the Web. *Journal of Web Engineering*, 4(1):165–186, 2005.

[15] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *TREC 1994*.

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

[17] R. L. T. Santos, C. Macdonald, and I. Ounis. Selectively diversifying Web search results. In *Proceedings of CIKM 2010*.

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

[19] H. Zaragoza, N. Craswell, M. J. Taylor, S. Saria, and S. E. Robertson. Microsoft Cambridge at TREC 13: Web and Hard tracks. In *Proceedings of TREC 2004*.