University of Glasgow at TREC 2013: Experiments with Terrier in Contextual Suggestion, Temporal Summarisation and Web Tracks

Richard McCreadie, M-Dyaa Albakour, Stuart Mackie, Nut Limosopathan Craig Macdonald and Iadh Ounis
{firstname.lastname}@glasgow.ac.uk
School of Computing Science
University of Glasgow
Glasgow, UK

B. Taner Dinçer*
dtaner@mu.edu.tr
Dept of Statistics & Computer Engineering
Mugla University
Mugla, Turkey

ABSTRACT

In TREC 2013, we focus on tackling the challenges posed by the new Contextual Suggestion and Temporal summarisation tracks, as well as enhancing our existing technologies to tackle the new risk-sensitive aspect of the Web track, building upon our Terrier Information Retrieval Platform. In particular, for the Contextual Suggestion track, we investigate how to exploit location-based social networks, with the aim of better identifying venues within a city that a given user might be interested in visiting. For the Temporal Summarisation track, we propose a new summarisation framework and investigate novel techniques to adaptively alter the summarisation strategy over time. For the TREC Web track, we continue to build upon our learning-to-rank approaches and novel xQuAD / Fat frameworks within Terrier, increasing effectiveness when ranking and examining two new approaches to risk-sensitive retrieval.

1. INTRODUCTION

In TREC 2013, we participate in Web adhoc and risk-sensitive tasks, the Contextual Suggestion track "entertain me" task and the Temporal Summarisation sequential update summarisation task. Our focus is the development of effective and efficient approaches to these tasks, building upon our open-source Terrier Information Retrieval (IR) platform [16]. Indeed, our Web track participation focuses on further developing the core data-driven ranking models and infrastructure within Terrier, in-line with the Terrier vision [9]. Meanwhile, our Contextual Suggestion and Temporal Summarisation participations revolve around the development of new real-time streaming applications and technologies building upon Terrier.

In the Contextual Suggestion track, our aim is to develop effective approaches to identify venues that a user with a given past history might be interested in visiting within a city. This task is challenging, since without an explicit representation of the user's current information need, the potential interests of the user need to be inferred from the sparse user profile. We propose a novel approach to tackle contextual suggestion that exploits implicit knowledge within freely available location-based social networks regarding the popularity of venues, as well as venue density information to better identify the currently 'hot' venues in a city that match the user's profile. Furthermore, we also investigate a new approach that uses an explicit diversification strategy to increase the coverage of venue types in the top of the venue ranking suggested.

We also participate in the sequential update summarisation task of the Temporal Summarisation track. The major goal of our participation is to develop effective incremental summarisation approaches for a given event. To this end, we propose a new summarisation framework that combines both effective document search approaches within Terrier with state-of-the-art summarisation techniques to produce extractive summaries that update over time. Using an implementation of this framework within the Storm distributed stream processing framework,[1] we developed a wide variety of summarisation strategies optimised for different conciseness, cohesiveness and diversity scenarios. Moreover, we also proposed and deployed two novel adaptive content selection techniques that use topic modelling adaptively alter the summarisation strategy over time to minimise topic drift.

In our participation in the Web track, our primary goal is to enhance our data-driven learning infrastructure within Terrier for use on the new ClueWeb12 corpus. In particular, for the adhoc ranking task, we deploy our state-of-the-art xQuAD / Fat frameworks within Terrier using a variety of relevance, authority, quality and spam features. For the risk-sensitive task, we employ a novel risk-sensitive learning to rank algorithm and a new approach that selectively applies one of a set of document ranking models based upon an estimate of their predicted riskiness for the current query.

The remainder of this paper is structured as follows. In Section 2, we describe our participation in the Contextual Suggestion track. Section 3 details our participation in the new Temporal Summarisation track. In Section 4, we describe our Web track adhoc and risk-sensitive task participations. Conclusions are provided in Section 5.

2. CONTEXTUAL SUGGESTION TRACK

The main aim of our participation in the TREC 2013 Contextual Suggestion track is to extend and refine novel contextual retrieval models, which we have developed upon our Terrier IR platform to address emerging information needs

*Work conducted while visiting the University of Glasgow.

[1] http://storm-project.net

in smart cities, such as the "entertain me" zero-queries tackled in this track.

The emergence of Location-based Social Networks (LSNs) such as FourSquare and Facebook Places offer enormous information that can be exploited to address the contextual recommendation problem in smart cities. Our approach aims at exploiting both the social aspect of the users in these networks, and the rich structured information available about the venues covered by these networks. We employ both aspects to effectively recommend to users venues to visit, without issuing a query. This is achieved by mining the implicit context of the users inferred from their location and the explicit interests in their profile.

To produce a personalised ranking of the venues for a given user, we build textual representations of both the user's profile and the venues available in the LSN. Using a vector representation, we construct a profile of the user from the available explicit judgements. We take the 5-rating scores (0 to 4) given by the user in the track dataset and convert them into positive and negative judgements that are then incorporated into the vector representation. For the venues, we use the home page of the venue on the LSN as a textual representation. The home page of the venue contains information about the venue such as its name, its description and the category of that venue. In addition, it incorporates a social aspect in the form of the comments provided by users. Such information enriches the vector representations of the venue. Generally, our approach aims at computing the textual similarity between the profile of the user and the venues close to the user. Using this similarity score, we can rank the venues and recommend them to the user. We then incorporate other features available about the venues from the LSN as follows:

First, we introduce a social aspect to the venue ranking by integrating an estimation of the popularity of the venue as obtained from the previous interactions of the users on the LSNs. In our estimation of the popularity, we take into account the fact that the size and the population of an area or a city may affect the volume of the user activity on the LSNs. Therefore, we normalise the popularity estimate by considering all the venues in the surrounding areas, such that venues with lower volume of LSN activity within less populated areas are boosted and vice versa.

Moreover, we recognise that a zero-query is ambiguous by definition. Hence, inspired by diversification approaches in web search to address ambiguous queries, we develop and deploy a personalisation model based on the xQuAD diversification framework [20]. Our model personalises the recommended venues to cover the categories of interest for each user - these categories of interest are inferred automatically from the user's profile.

Using the Foursquare LSN, we crawl venues for the various contexts (cities) used in the track. Using these venues, we devised three different runs to evaluate our approach described above (uogTrCF, uogTrCFX and uogTrCFP). Only the last two were submitted:

- **uogTrCF**: This run serves as our baseline. Venues for each user profile and context pair are ranked using the similarity score between the user profile and the venue.

- **uogTrCFP**: This run investigates the usefulness of using the social popularity feature to inform the selection of venues. It uses a linear combination between the similarity and the normalised popularity of the venue estimated. The normalised popularity of the venue is estimated by using the volume of the user population visits (FourSquare "checkin"s) and taking into consideration the overall volume of the user population visits in the surrounding area.

- **uogTrCFX**: This run re-ranks venues in order to cover diverse categories of the user interests using the xQuAD framework as described above. The categories used are those available on the venue's profile on FourSquare. In particular, we use the top level category from the hierarchy of venues' categories provided in FourSquare.

	Submitted	P(5)	MRR	TBG
TREC Median	-	0.2368	0.3415	0.8593
uogTrCF	✖	0.2170	0.4170	-
uogTrCFX	✔	0.2332	0.4022	1.0894
uogTrCFP	✔	**0.2753**	**0.4327**	**1.3568**

Table 1: Results of our runs in the Contextual Suggestions track. Figures in bold represent the top performances. Note that the TBG cannot be estimated from the relevance assessments.

Table 1 reports the performance of our two submitted runs and the non-submitted run together with the TREC Median using the official measures. First, we observe that our submitted runs achieve above median performance for all measures (with the exception of P@5 for the uogTrCFX run, which provides equivalent performance). In particular, the uogTrCFP run, which incorporates the social popularity, achieves the best performance. This highlights the importance of the venue popularity signal when recommending places that a user might wish to visit. Our diversification run (uogTrCFX) that attempts to increase the number of venue categories appearing the the top ranks is also promising as it outperforms the baseline (uogTrCF), which does not consider diversification. However, we need to investigate more elaborate techniques when mapping between the user interests and a finer-grained category of the venue. For example, a better personalisation approach should differentiate between various types of cuisines, instead of targeting all the restaurants.

3. TEMPORAL SUMMARISATION TRACK

The aim of our participation in the first year of the Temporal Summarisation track is to investigate real-time extractive models for the summarisation of events from across multiple streams. For our participation in the sequential update summarisation task, we extend the real-time search capabilities of the Terrier IR platform [16] to facilitate the incremental extractive summarisation and tracking of search topics in an extensible manner. In this way, we combine the effectiveness of state-of-the-art search techniques for finding relevant content with incremental summarisation strategies to filter down to a concise description of each event that can be updated over time. Furthermore, we investigate novel approaches to adaptively re-adjust the summarisation strategy over time with respect to the topic's prominence and the novelty of content available, as a means to tackle topic drift and reduce verbosity in the resultant summarisation.

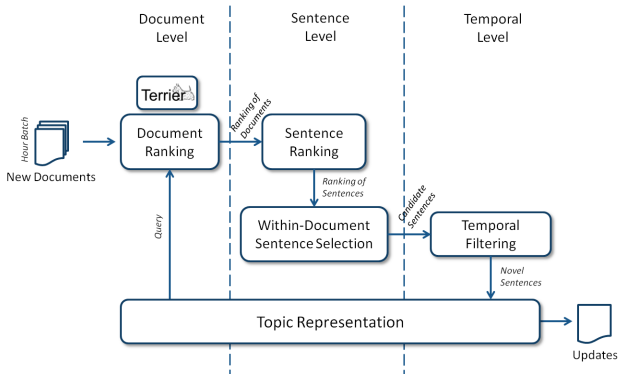

Figure 1: Overview of our core summarisation framework.

To perform summarisation, we define a *core summarisation framework* that enables us to examine different summarisation strategies. Under this framework, new documents are processed in temporal batches of one hour, resulting in zero or more sentences from those batches being emitted for the topic for that hour. Figure 1 illustrates how an hour batch of documents is processed. In particular, each document batch is processed by three levels, namely: document; sentence and temporal. At the document level, the new batch of documents representing the current hour is indexed by the Terrier instance and then the top 10 documents are ranked by their relatedness to the topic representation (query), producing a ranked list of the most related content from within the streams from the last hour. At the sentence level, the sentences within each of the 10 ranked documents are then ranked using a second target criteria (relatedness to the document front-matter, topical relevance or document-level salience), identifying the sentences that are most likely to be useful for inclusion. Next, a selection strategy is applied to the ranked sentences, filtering out redundant sentences in a greedy manner. The remaining candidate sentences are then passed to the temporal level to be compared against the current representation of the event. Any novel content that was not selected from prior batches is both emitted as updates and is also used to enrich the topic representation.

Using the core summarisation framework, we deploy three different extractive summarisation strategies by employing different techniques at each level of the framework, resulting in three different runs (uogTrNSQ1, uogTrNMM and uogTrEMMQ2)

- **uogTrNSQ1**: A precision-orientated run that focuses on filtering at the sentence level. In particular, it selects only most related sentence to the topic from each hourly batch.
- **uogTrNMM**: Leverages a more recall-orientated multi-document summarisation strategy, using the Maximal-Marginal Relevance (MMR) [3] algorithm to select novel content from each ranked document each hour.
- **uogTrEMMQ2**: Uses topic expansion from a timely Wikipedia corpus and WordNet at the sentence level to more accurately identify sentences related to the topic.

Run	Precision/Recall Orientated	Adaptive?	Expected Latency Gain	Latency Comprehensiveness
TREC Average	N/A	N/A	0.0599	**0.2996**
uogTrNSQ1	Precision	✖	0.0603	0.1844
uogTrNMM	Recall	✖	0.0452	0.2535
uogTrEMMQ2	Recall	✖	0.0396	0.2587
uogTrNMTm1MM3	Precision	✔	**0.0694**	0.2158
uogTrNMTm3FMM4	Recall	✔	0.0488	0.1704

Table 2: Performance of our submitted runs to the sequential update summarisation task.

However, within each of the data streams, we observed a high variance in a topic's prominence as it evolves over time. As a result, during periods of low prominence, weakly-related or off-topic content is likely to be summarised, leading to topic drift within the final summary. Hence, there is a need to identify when to avoid incorporating off-topic content into the summaries. To tackle this issue, we proposed and deployed two novel adaptive content selection techniques that use topic modelling at the document level to gauge topic prominence for a period of time, allowing the readjustment of the volume of content to be summarised when novel updates are less frequent, resulting in two further runs (uogTrNMTm1MM3 and uogTrNMTm3FMM4):

- **uogTrNMTm1MM3**: Uses a three-state automaton (no-content, limited-content and bursting) to determine the volume of content to select from each hourly batch. The automaton state is determined via overlap between the given topic representation and a set of topical areas generated by Gibb's sampling over the top documents ranked.
- **uogTrNMTm3FMM4**: Uses a more recall-focused adaptive technique. Computes the degree of overlap between the given topic representation and topical areas generated by Gibb's sampling and uses it to estimate the amount of content to select. This run also leverages a post-summarisation filtering technique based on a combination of topical relevance and sub-stream priors (news vs. social, vs. forum) to increase the quality of the generated summaries.

Table 2 reports the performance of our five submitted runs in terms of expected latency gain and latency comprehensiveness. From our submitted runs, we observe the following points of interest. First, in terms of expected latency gain, precision-orientated runs outperform recall-orientated runs by a large margin, indicating that the recall orientated runs are being heavily penalised for returning many updates. This in turn indicates that, under the track measures, small summaries (<100 sentences) are preferable. Second, of the non-adaptive runs submitted, we see that the recall-orientated run using MMR provides the best compromise between expected latency gain and latency comprehensiveness, showing that within-document diversification is an important direction for future investigation. Meanwhile, the lower performance of uogTrEMMQ2 in comparison to uogTrNMM indicates that the topic expansion strategy tested that leverages Wikipedia and WordNet was not effective. This is because the expansion process caused topic drift in the summaries produced. Finally, comparing the adaptive runs to the non-adaptive runs, we see that both adaptive runs outperform all of the non-adaptive runs submitted in terms of expected latency gain, indicating that adapting the sentence selection strategy over time is critical

for effective temporal update summarisation. Indeed, the adaptive uogTrNMTm1MM3 run outperforms the average of TREC systems under expected latency gain.

Overall, we conclude that the core summarisation framework that we proposed can be effective for sequential update summarisation. However, the techniques employed at each layer should focus on increasing precision due to the high levels of redundancy in the corpus. Moreover, as illustrated by our adaptive summarisation runs, altering the sentence selection strategy over time is a promising area to improve summarisation effectiveness.

4. WEB TRACK

In our participation to the adhoc and risk-sensitive tasks of Web track, we have two aims. First, to enhance and assess the performance of our data-driven learning infrastructure [10] that has proven effective during previous participations [8, 12, 13, 21] for the more recent ClueWeb12 corpus. Second, to investigate approaches to risk-aware retrieval. To this end, we begin by investigating learning to rank approaches within Terrier using our *fat framework* [11] for the fast computation of document features. Similar to past TREC participations [13, 8], we train upon ClueWeb09. We then propose and examine two new approaches to minimise risk-sensitivity within a learning environment, based on risk-sensitive learning to rank [22] and the predictive selection of retrieval models per-query using estimated risk.

We index category A (∼716M English documents) and category B (∼50M English documents) subsets of the ClueWeb12 corpus without stemming or stopwords. At retrieval time, we apply one of several retrieval models (DPH from the Divergence from Randomness framework [1], DFIC from the Divergence from Independence framework [6] or BM25) to identify the *sample* documents to re-rank using the learned models. Following the recommendations of [11] for ClueWeb09, we select the top 5000 documents for re-ranking using learning to rank, where the weighting model does not consider anchor text.

For applying learning to rank, our category A and B runs both use a total of 63 features, as described in Table 3. Note that many different weighting model features are computed, as they can contribute differently to the learned models [11]. We also observe that there is no need to train the hyperparameters of those weighting models that typically control document length normalisation, as the learning to rank technique will implicitly address any bias towards short or long documents as part of its learning process [11].

The same features are computed on ClueWeb09 queries for the purposes of training. We thereafter deploy two learning to rank techniques, namely AFS [14] – which creates a linear learned model – and also the state-of-the-art LambdaMART learning to rank technique [7, 23],[2] which creates a learned model based on regression trees. To train the learning to rank techniques, we use 200 queries from the the TREC Web tracks 2009-2012, randomly split into training and validation sets, so as to prevent overfitting.

Moreover, we tested the sensitivity of the learned models wrt. the document weighting model that is used to generate the initial ranking of documents, by contrasting new ranking models from the Divergence from Independence (DFI) and Divergence from Randomness (DFR) families.

[2]`http://code.google.com/p/jforests/`

Next, for the purposes of the risk-sensitive retrieval task, we experimented with two techniques for reducing risk during retrieval: In particular, through a thorough statistical analysis of 115 features that are calculated for each query, we trained a novel selection technique that aimed to select the most effective/safe retrieval strategies for a given query; We also investigated the adaptation of a learning to rank technique that makes it inherently sensitive to risk when learning a ranking model, also known as $\mathcal{U}_{RISK}$ [22].

We submitted six runs to the adhoc and risk-sensitive retrieval tasks of the Web track, covering both category A and category B of the ClueWeb12 corpus, and deploying 63 features on both corpora for the purposes of learning to rank. On category A: uogTrAIwLmb combines DFI and a regression trees-based learning to rank technique; uogTrADnLrb uses instead a DFR model and a risk-sensitive learning to rank technique; uogTrAS1Lb and uogTrAS2Lb are selective approaches, using different learned models on a per-query basis; Finally, on category B, uogTrBDnLaxw and uogTrBDnLmxw deploy a DFR model and our existing effective xQuAD diversification framework [19], differing only in the type of learning to rank technique deployed, namely linear vs. regression trees. Table 4 summarises the configuration of each of six submitted runs, as well as 6 unsubmitted runs that we also evaluate.

Table 5 reports the effectiveness of all six of our submitted Web track runs[3], as well as various unsubmitted runs, and the four provided standard baselines. Results are reported in terms of NDCG@20 and ERR@20, as well as risk-aware $\mathcal{U}_{RISK}$ variants for $\alpha = 1$ and $\alpha = 5$, compared to the Indri query likelihood unfiltered standard baseline for the respective category.[4] Firstly, on analysing the submitted runs based on the category B subset of ClueWeb12, we have the following observations:

- In contrast to previous results on the ClueWeb09 corpus [4, 18], category B of ClueWeb12 provided overall lower performance than category A. This can be attributed to a much lower number of relevant documents per-topic found in the category B subset of ClueWeb12 (approx. 19%), compared to ClueWeb09 which was used for training (approx. 49% for TREC 2009 [18]).

- Comparing uogTrBDnLaw with uogTrBDnLmw, and uogTrBDnLaxw with uogTrBDnLmxw, we find that the linear AFS learning to rank techniques gives generally higher effectiveness (one exception for xQuAD according to ERR@20).

- The xQuAD diversification framework always improves adhoc effectiveness: uogTrBDnLaxw vs. uogTrBDnLaw, uogTrBDnLmxw vs. uogTrBDnLmw. This is in line with our previous observations for ClueWeb09 [8].

- Finally, for each adhoc effectiveness measure, the most effective run is also the most risk-averse compared to the Indri standard baseline run, according to the corresponding $\mathcal{U}_{RISK}$ measure.

[3]We report revised scores for several runs, based on a corrected implementation of risk-aware LambdaMART.

[4]We were unable to produce the risk-aware evaluation results provided by NIST, and as such, we do not report the TREC median for these measures.

Features	Total
Sample: DPH, DFIC or BM25	1
Weighting models on the whole document [11] (DFRee, DPH [1], PL2 [1], BM25, Dirichlet LM, MQT [10], LGD, DFIC [6], DFIZ [6])	8
Weighting models as above on each field, namely: title, URL, body and anchor text; + PL2F	37
Term-dependence proximity models (MRF [15], pBiL [17])	2
URL (e.g. length) link (e.g. inlink counts) & content quality (e.g., fraction of stopwords, table text [2], spam classification [5]) features	15
TOTAL	63

Table 3: Document features used in the Web track, both Category A and Category B runs.

ID	Submitted	Category	Stemming	Sample	LTR	Other
uogTrAIwLab	✖	A	Weak	DFIC	AFS	-
uogTrAIwLmb	Adhoc	A	Weak	DFIC	LambdaMART	-
uogTrADnLrb	Risk	A	None	DPH	Risk-aware LambdaMART	-
uogTrADnLmb	✖	A	None	DPH	LambdaMART	-
uogTrABwLab	✖	A	Weak	BM25	AFS	-
uogTrABwLmb	✖	A	Weak	BM25	LambdaMART	-
uogTrAS1Lb	Risk	A	-	-	-	Selective (uogTrABwLab/uogTrADnLrb)
uogTrAS2Lb	Adhoc	A	-	-	-	Selective (uogTrABwLab/uogTrADnLrb/uogTrAIwLmb)
uogTrBDnLaw	✖	B	None	DPH	AFS	-
uogTrBDnLmw	✖	B	None	DPH	LambdaMART	-
uogTrBDnLaxw	Risk	B	None	DPH	AFS	xQuAD
uogTrBDnLmxw	Adhoc	B	None	DPH	LambdaMART	xQuAD

Table 4: Summary of submitted and unsubmitted runs to the adhoc and risk-sensitive tasks of the Web track.

Next, on analysing the category A runs, we obtain the following observations:

- Our data-driven learning to rank approaches were all substantially above the both the TREC median performances, and the standard Indri baselines provided by the organisers.
- Our most effective run, uogTrAIwLmb, deployed DFIC, weak stemming and LambdaMART.
- Next, comparing the learning to rank techniques, we find no clear winner for AFS vs. LambdaMART: uogTrAIwLmb (LambdaMART) is more effective than uogTrAIwLab (AFS), but uogTrABwLab is more effective than uogTrABwLmb.
- Comparing uogTrADnLrb and uogTrADnLmb, we observe that risk-aware LambdaMART using $\mathcal{U}_{RISK}$ can slightly improve adhoc ERR@20 compared to normal LambdaMART (at the cost of marginal NDCG@20 loss). This combination also markedly reduces risk, both for NDCG@20 and ERR@20. Indeed, for ERR@20, uogTrADnLrb exhibits the best $\mathcal{U}_{RISK}$ performance across all of our runs.
- Our selective approaches, uogTrAS1Lb and uogTrAS2Lb (which selects between two runs and three runs, respectively), are very similar in effectiveness. Their observed effectiveness' intersect the performance of their respective constituent runs.
- Finally, in line with our category B observations, our most effective category A run (uogTrAIwLmb) is also the most risk-averse, according to the $\mathcal{U}_{RISK}$ measures.

Overall, we conclude that with performances substantively about track median, our general data-driven approach based on learning to rank for Web search is effective. Diversification remains an excellent technique to enhance adhoc effectiveness. Finally, compared to standard query likelihood Indri baseline runs, we find that risk-averseness is correlated with effectiveness, but that a risk-aware version of LambdaMART can reduce the amount of risk observed.

5. CONCLUSIONS

In TREC 2013, we participated in the Web adhoc and risk-sensitive tasks, the Contextual Suggestion track "entertain me" task and the Temporal Summarisation sequential update summarisation task, building upon our Terrier IR platform. In particular, for the Web track, we leveraged data-driven learning using our state-of-the-art xQuAD and Fat frameworks, markedly outperforming the median of TREC systems, as well as investigating new machine learning and per-query selective approaches to minimise risk when ranking. For the Contextual Suggestion track, we proposed a novel approach that leverages localised popularity and density estimations from location-based social networks to better suggest currently 'hot' venues for the user. Finally, for the Temporal Summarisation track, we proposed a new new summarisation framework that combines both effective search approaches with state-of-the-art summarisation to produce extractive summaries that update over time and examined new adaptive techniques to model how to select content as events evolve.

6. 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 *Proc. of TREC*, 2007.

[2] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of Web documents. In *Proc. of WSDM*, 2011.

Run	Submitted Task	Category	Adhoc		$\mathcal{U}_{RISK}$ $\alpha = 1$		$\mathcal{U}_{RISK}$ $\alpha = 5$	
			NDCG@20	ERR@20	NDCG@20	ERR@20	NDCG@20	ERR@20
TREC median	-	A	0.1739	0.0980	-	-	-	-
Indri QL	Std	A	0.1803	0.0989	0	0	0	0
Indri QL Spam filtered	Std	A	0.1721	0.0931	-0.0671	-0.0429	-0.3027	-0.1908
Indri RM	Std	A	0.1641	0.0884	-0.0404	-0.0290	-0.1369	-0.1027
Indri RM Spam filtered	Std	A	0.1682	0.0963	-0.0713	-0.0389	-0.3078	-0.1836
uogTrAIwLab	✖	A	0.1975	0.1384	-0.0298	0.0110	-0.2175	-0.1027
uogTrAIwLmb	Adhoc	A	**0.2594**	**0.1604**	**0.0494**	0.0359	**-0.0693**	-0.0662
uogTrABwLab	✖	A	0.2300	0.1556	0.0128	**0.0366**	-0.1349	-0.0434
uogTrABwLmb	✖	A	0.2194	0.1442	-0.0037	0.0237	-0.1748	-0.0628
uogTrADnLrb*	Risk	A	0.2419	0.1516	0.0336	0.0358	-0.0782	**-0.0315**
uogTrADnLmb	✖	A	0.2471	0.1415	0.0366	0.0256	-0.0843	-0.0425
uogTrAS1Lb*	Risk	A	0.2262	0.1542	0.0058	0.0349	-0.1546	-0.0465
uogTrAS2Lb*	Adhoc	A	0.2396	0.1551	0.0234	0.0334	-0.1201	-0.0574
Indri QL	Std	B	0.1066	0.0749	0	0	0	0
Indri QL Spam filtered	Std	B	0.0908	0.0669	-0.0570	-0.0405	-0.2212	-0.1704
Indri RM	Std	B	0.0825	0.0472	-0.0521	-0.0565	-0.1641	-0.1716
Indri RM Spam filtered	Std	B	0.0886	0.0633	-0.0613	-0.0457	-0.2340	-0.1825
uogTrBDnLaw	✖	B	0.1565	0.1021	0.0422	0.0123	0.0117	-0.0475
uogTrBDnLmw	✖	B	0.1350	0.0955	0.0128	0.0007	-0.0495	-0.0791
uogTrBDnLaxw	Risk	B	**0.1772**	0.1131	**0.0659**	0.0260	**0.0476**	-0.0229
uogTrBDnLmxw	Adhoc	B	0.1708	**0.1203**	0.0587	**0.0335**	0.0365	**-0.0143**

Table 5: Results of our submitted and unsubmitted runs for the Web track under the normalised discounted cumulative gain at rank 20 (NDCG@20) and expected reciprocal rank at rank 20 (ERR@20) measures, as well as $\mathcal{U}_{RISK}$ equivalents for $\alpha = 1$ and $\alpha = 5$. * denotes corrected results.

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

[4] C. L. A. Clarke, N. Craswell, I. Soboroff, and G. V. Cormack. Overview of the TREC 2010 Web track. In *Proc. of TREC*, 2010.

[5] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large Web datasets. *Information Retrieval*, 14(5):441–465, 2011.

[6] B. T. Dinçer, I. Kocabas, and B. Karaoglan. IRRA at TREC 2010: Index term weighting by divergence from independence model. In *Proc. of TREC*, 2010.

[7] Y. Ganjisaffar, R. Caruana, and C. Lopes. Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proc.s of SIGIR*, 2011.

[8] N. Limsopatham, R. McCreadie, M.-D. Albakour, C. Macdonald, R. L. T. Santos, and I. Ounis. University of Glasgow at TREC 2012: Experiments with Terrier in Medical Records, Microblog, and Web tracks. In *Proc. of TREC*, 2012.

[9] C. Macdonald, R. McCreadie, R. L. Santos, and I. Ounis. From Puppy to Maturity: Experiences in developing Terrier. *Proc. of OSIR at SIGIR*, 2012.

[10] C. Macdonald, R. Santos, and I. Ounis. The whens and hows of learning to rank for Web search. *Information Retrieval*, 16(5):1–45, 2012.

[11] C. Macdonald, R. L. Santos, I. Ounis, and B. He. About learning models with multiple query-dependent features. *ACM Transactions on Information Systems*, 31(3):1–11, 2013.

[12] R. McCreadie, C. Macdonald, I. Ounis, J. Peng, and R. L. T. Santos. University of Glasgow at TREC 2009: Experiments with Terrier—Blog, Entity, Million Query, Relevance Feedback, and Web tracks. In *Proc. of TREC*, 2009.

[13] R. McCreadie, C. Macdonald, R. L. T. Santos, and I. Ounis. University of Glasgow at TREC 2011: Experiments with Terrier in Crowdsourcing, Microblog, and Web tracks. In *Proc. of TREC*, 2011.

[14] D. Metzler. Automatic feature selection in the Markov random field model for Information Retrieval. In *Proc. of CIKM*, 2007.

[15] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proc. of SIGIR*, 2005.

[16] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance and scalable Information Retrieval platform. In *Proc. of OSIR at SIGIR*, 2006.

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

[18] R. L. Santos, C. Macdonald, and I. Ounis. Effectiveness beyond the first crawl tier. In *Proc. of CIKM*, 2011.

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

[20] R. L. T. Santos, C. Macdonald, and I. Ounis. Intent-aware search result diversification. In *Proc. of SIGIR*, 2011.

[21] R. L. T. Santos, R. McCreadie, C. Macdonald, and I. Ounis. University of Glasgow at TREC 2010: Experiments with Terrier in Blog and Web tracks. In *Proc. of TREC*, 2010.

[22] L. Wang, P. N. Bennett, and K. Collins-Thompson. Robust ranking models via risk-sensitive optimization. In *Proc. of SIGIR*, 2012.

[23] Q. Wu, C. J. C. Burges, K. M. Svore, and J. Gao. Ranking, boosting, and model adaptation. Technical Report MSR-TR-2008-109, Microsoft, 2008.