University of Glasgow at TREC 2015: Experiments with Terrier in Contextual Suggestion, Temporal Summarisation and Dynamic Domain Tracks

Richard McCreadie[1], Stuart Mackie[2], Jarana Manotumruksa[3],
Graham McDonald[4], Saúl Vargas[1], M-Dyaa Albakour[1], Craig Macdonald[1],
and Iadh Ounis[1]
[1]{firstname.lastname}@glasgow.ac.uk
{[2]s.mackie.1,[3]j.manotumruksa.1,[4]g.mcdonald.1}@research.gla.ac.uk
School of Computing Science
University of Glasgow
Glasgow, UK

ABSTRACT

In TREC 2015, we focus on tackling the challenges posed by the Contextual Suggestion, Temporal Summarisation and Dynamic Domain tracks. For Contextual Suggestion, we investigate the use of user-generated data in location-based social networks (LBSN) to suggest venues. For Temporal Summarisation, we examine features for event summarisation that explicitly model the entities involved in the events. Meanwhile, for the Dynamic Domain track, we explore resource selection techniques for identifying the domain of interest and diversifying sub-topic intents.

1. INTRODUCTION

In TREC 2015, we participate in both the "live" and "batch" experiments of the Contextual Suggestion track, the Summarisation Only task (Task 3) of the Temporal Summarisation track and the main task of the Dynamic Domain track. Our focus is the development of effective and efficient approaches to these tasks, building upon our open-source Terrier Information Retrieval (IR) platform [9] and extensive experience working with machine learned models [10].

In the Contextual Suggestion track, we leverage data from the Foursquare location-based social network (LBSN) to suggest venues to users. In particular, we propose two novel venue suggestion approaches, based on factorisation machines and a context-aware learning to rank technique, respectively. These approaches both use Foursquare data as well as the contextual information about the user to suggest venues. The factorisation machine-based approach uses check-in statistics and venue categories together with the context of the user to produce personalised suggestions. Meanwhile, our learning to rank approach incorporates two main components: a component containing user and venue-dependent features combined using the LambdaMART learning to rank technique; and a component that uses probabilistic time and term-based approaches to predict the contextual appropriateness of venues.

We also participate in the Temporal Summarisation track, specifically Task 3 "Summarisation Only" using the "RelOnly" corpus. The aim of our participation is to investigate algorithms for the summarisation of events, explicitly modelling the entities involved in an event, and the interaction between such entities. We propose and evaluate features based on estimates of entity importance and entity-to-entity interactions, where the estimates are derived from the input document stream. Furthermore, we also investigate the effectiveness/latency trade-off within the task, by testing two methods for processing the corpus, namely: by streaming over each event timespan and summarising document-by-document; or batching documents in hourly chunks to be summarised.

Finally, for our participation in the Dynamic Domain track, we aim to investigate methods for minimising the number iterations of the retrieval-feedback cycle needed to correctly identify the sub-topics that are of interest to the user. To do this, we first view the task as a resource selection problem and experiment with resource selection and document prioritisation techniques for identifying the domain of interest for each topic. Secondly, we investigate search result diversification as a means to increase the number of potential sub-topic intents shown to the user within each iteration, thereby maximising the user's potential exploration rate.

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 Temporal Summarisation track. In Section 4, we describe our participation in the main task of the Dynamic Domain track. Conclusions are provided in Section 5.

2. CONTEXTUAL SUGGESTION TRACK

Similar to previous years, for TREC 2015, the Contextual Suggestion track asks participants to suggest a ranked list of venues to users, based upon their profiles and preferred contexts. The track consists of both live and batch experiments. For the live experiment, the participants have to setup and register their system with the organisers, and respond in real-time to user's requests. Each request consists of the user's profile (expressed as ratings of a set of venues) and contextual preferences (e.g. duration of visit). The response contains a list of venue IDs in the collection provided by TREC. For the batch experiment, given a user profile and contextual preferences, participants were asked to rank sets of candidate venues suggested during the live experiment.

For TREC 2015, the user's context contains the city where the user is looking for venues to visit, as well as newly pro-

posed contextual preferences, namely: *duration* of trip (daytime, nighttime, weekend, longer), the *season* of the year (Spring, Summer, Autumn and Winter), the *group* of people the users are intending to visit the venue with (alone, friends, family and others) and the *type* of the trip (business, holiday and other). In the following, we describe our proposed approach for the live (Section 2.1) and batch experiments (Section 2.2). Finally, Section 2.3 highlights our submitted runs and their achieved performances.

2.1 Live Experiment

Our setup of the live experiment consisted of indexing venues from 272 cities, as listed in the Foursquare location-based social network (LBSN). Then, we registered three different systems that ranked venue suggestions in response to a user query. For the live experiment, the user contexts were limited in nature, in that users only rate venues from 2 different cities. For this reason, and to ensure that Foursquare venues were represented in the batch experiment, we use three unsupervised ranking approaches to contribute a wide number of appropriate Foursquare venues. Our three different systems for the live experiment were as follows:

- Venue-independent: a system that ranks venues based upon popularity, computed from the number of check-ins, photo and tips that each venue experiences on Foursquare.
- User-dependent: a system that ranks venues based on the Cosine similarity between the Foursquare categories of positively rated venues in the user's profile and the Foursquare categories of the venue itself.
- Contextual-preferences and User-dependent: a system that combines two aspects: (a) how well the venue matches the contextual preferences of the user (duration of visit, season, group, type of trip), computed based on the timestamps of photos of the venue from Foursquare; and (b) the *user-dependent* system described above. These two approaches are linearly combined with even weight.

2.2 Batch Experiment

In the batch experiment of the Contextual Suggestion track, we make use of two different learning approaches, as well as novel models for dealing with new aspects of the contextual preferences expressed by users. In particular, in the following we describe our factorisation machines and learning-to-rank approaches to both rank venues and predict the contextual appropriateness of a venue.

Note that while the batch experiment is intended to rank an initial set of venues, we make use of data from the Foursquare location-based social network, and therefore ignored venues that did not appear on Foursquare. More precisely, we assign any venue not appearing in Foursquare a zero score, and so appear at the bottom of our submitted rankings.

2.2.1 Learning to Rank

For each venue in the initial set, we compute *venue-dependent* and *user-dependent* features, as proposed by Deveaud *et al.* [3]. In particular, a total of 49 features are computed for each venue-user pair based on the Foursquare data, such as check-ins, users and tips counts and venue categories. Moreover, following the same experimental setup as we applied for TREC 2014 [12], we use the LambdaMART [18] learning to rank technique to learn an effective combination of those features to re-rank venues. We train the learner using the 2014 Contextual Suggestion dataset. The final score for an unseen venue v for a user u is denoted $score_{LTR}(v, u)$.

2.2.2 Factorisation Machines

Factorisation machines [14] are a generalisation of the well-known matrix factorisation techniques [6] that have been successfully used in the area of collaborative filtering-based Recommender Systems. Factorisation machines can leverage not only the feedback of the user for venues she previously visited, but also user-related, venue-related and contextual information. They are therefore an appropriate approach for this track.

Factorisation machines receive as input instances that enclose the information related to a user, a venue he/she visited and the context of the visit in the form of numerical vectors. In our case, our instances are comprised of three blocks representing the following information: 1) a user indicator block to enable personalisation, 2) the user features and context provided in the batch requests and 3) venue-dependent features extracted from Foursquare, as described above. Note that, unlike a collaborative filtering setting where the suggested venues have previously been visited by at least one user, we do not use an item indicator block for our instances.

With the previous format for the instances, we train our factorisation machines to reduce the error in the ranking of the user profiles provided in the batch requests data. Specifically, we have adapted the list-wise error function of ListRank [16] for our factorisation machine model. The optimal parameters of the model (learning rate, regularisation parameters) were determined with a train/test partition of the batch requests data, where the ratings expressed by users on venues from the two seed cities (Seattle and Detroit) were used for training the models, and the remainder of the venues were used for evaluation.

2.2.3 Predicting Contextual Appropriateness

A major novelty in the setup of the TREC 2015 Contextual Suggestion track is the introduction of contextual aspects that can be expressed by users, namely: *duration of visit* (daytime, nighttime, weekend, longer); *season* (Spring, Summer, etc.), *group* (alone, with friends, with family, other), and *type of trip* (business, holiday, other). It is therefore important to take these contextual aspects into account when ranking venues.

In our participation, we developed different approaches to accomplish appropriate prediction of the contextual aspects of the venues, using timestamp and textual information that can be gleaned about the venues from the Foursquare location-based social network, and the venues' websites.

- **Timestamp**: For the duration and season dimensions, we propose the use of timestamp information that can be gleaned about the venues from the Foursquare location-based social network. In particular, we observe that the timing of photos of a venue uploaded to the LBSN can be indicative of appropriateness in terms of duration and season dimensions. For a venue v and a contextual dimension $di \in A$ for aspect $A = \{duration, season\}$ expressed by user u, we score the denote our the predicted contextually appropriateness as $d_A(v, di)$.

	Submitted	P@5	MRR
TREC Median	-	0.5090	0.6716
Foursquare baseline	✖	0.5100	0.6509
uogTrCSFM	✔	**0.5706**	**0.7190**
uogTrCSLVPC	✔	0.5498	0.6758

Table 1: Results of our runs in the Contextual Suggestions track. Figures in bold represent the top performances.

- **Textual**: For the group and type of trip contextual aspects, we use Terrier to index the websites of each venue. Then, given a contextual aspect dimension, we identify lists of terms related to that dimension, based on freely available Web resources. For instance, for the family dimension of the group aspect, our list had terms such as "brother", "mother", etc. We then score venue websites, based on mentions of dimension related terms, using the BM25 weighting model, i.e. for venue v with website v_w for a dimension $di \in A$ for aspect $A = \{group, type\}$, calculate $d_A(v, d) = score_{BM25}(v_w, Q_{di})$, where Q_{di} is the set of terms identified as related to dimension di.

2.3 Batch Experiment Runs & Results

We submitted 2 runs to the Contextual Suggestion Batch Experiment:

- uogTrCSFM deploys factorisation machines (see Section 2.2.2).
- uogTrCSLVPC deploys our learning to rank approach (Section 2.2.1) and the prediction of contextual appropriateness (Section 2.2.3) in a linear combination.

Table 1 reports the performance of our two submitted runs together with the TREC Median using the official measures. For reference, we include a baseline run consisting in ranking first the venues in Foursquare by venue id (denoted Foursquare baseline). Note that the baseline itself – which our runs are based on – fares well with respect to the TREC median, particularly in terms of P@5. As the results show, our runs are competitive, performing above the baseline and the TREC median in both cases. In particular, the *uogTrCSFM* run (factorisation machines) is the best of the two. Overall, the results for both of our runs exhibit promising above-median performances, and hence merit further study in the future.

3. TEMPORAL SUMMARISATION TRACK

The aim of our participation in the Temporal Summarisation track is to explore entity-focused models for the summarisation of evolving events [4]. In particular, we form the hypothesis that *events are about entities*, and effective summaries of events can be produced using summarisation features that are derived from the entities involved in the events. The features we investigate are entity importance and entity–entity interaction, which attempt to capture the salient entities and how they connect with other entities. Entity importance is estimated via entity frequency, and entity–entity interaction is estimated via entity co-occurrence. The entity-focused features are used in event summarisation algorithms to score sentences for inclusion into the summary of the event. Further, we also investigated two distinct methods of processing the corpus, summarising the content of each event either document-by-document, or in hour-by-hour batches. In the case of hour-by-hour, all sentences from documents within that hour are combined into a virtual document. Summarising each document as it arrives simulates a real-time scenario, whereas batching the documents in hourly chunks represents a near real-time task.

We submitted runs to Task 3 "Summarisation Only", using the "RelOnly" corpus, where the input to the event summarisation algorithm is a topically cohesive set of documents about an event. This contrasts with Task 1 and Task 2, and our participation in previous years [11, 12], where participants are required to perform some form of topic detection and tracking [1] over a larger corpus containing non-relevant documents, in addition to summarisation [5, 13]. Documents in the relevant only corpus[1] were first converted to plain text TREC <DOC>'s, discarding the binary encoded metadata. Then, we used the CoreNLP toolkit to tokenise the (Serif) pre-tagged sentences within the documents, and identify three classes (<PER>, <ORG>, <LOC>) of named entities. We compute entity frequency at the document (or virtual document) level, and compute entity co-occurrence at the sentence level.

To produce temporal summaries of events, first, the topic query is used to produce an initial ranking of sentences. Sentences are scored by their cosine similarity to the query, and sentences with no similarity to the query are discarded. This set of candidate summary sentences is then passed to an entity-focused event summarisation algorithm, for re-ranking. Further, we submitted (un-pooled) baseline runs where no re-ranking was performed. Next, at each batch boundary (document-by-document or hour-by-hour), all candidate summary sentences are scored using the entity-focused features, and passed through a top-k selection procedure, where $k = 1$ in our submitted runs. The selection of k sentences is then passed to an anti-redundancy filtering component, which aims to minimise repetition in the sentences being emitted over time. In our submitted runs, we use a cosine similarity threshold filter. Sentences passing this cosine similarity threshold are emitted from the system and form the summary of the event.

Table 2 and Table 3 presents the results of our runs, and the track average. Table 2 gives results for our runs that were pooled, and Table 3 gives extended evaluation results using automatic matching of non-pooled updates. Statistically significant differences from the TREC average are indicated in Table 2 and Table 3 using the "†" symbol, where the statistical test used is the Students t-test, paired-sample, with 95% confidence level. We submitted 6 runs in total, 4 of which were pooled, and 2 were un-pooled, as described below.

- Entity Importance: Scoring candidate summary sentences as a function of the entities they contain.
  - *uogTrdEQR1* (doc-by-doc)
  - *uogTrhEQR2* (hour-by-hour)
- Entity–entity Interaction: Scoring candidate summary sentences as a function of the entity pairs they contain.

[1] dcs.gla.ac.uk/~richardm/TREC-TS-2015RelOnly.aws.list

  - *uogTrdEEQR3* (doc-by-doc)
  - *uogTrhEEQR4* (hour-by-hour)
- Baselines (un-pooled): Scoring candidate summary sentences by their cosine similarity to the query.
  - *uogTrdSqCR5* (doc-by-doc)
  - *uogTrhSqCR6* (hour-by-hour)

Under the track target metric of the harmonic mean of normalised expected latency gain and latency comprehensiveness, from Table 2, we observe that all submitted runs performed above the track average. We also observe that processing the corpus using the hour-by-hour method is more effective than processing document-by-document, when selecting 1 update per batch boundary (top-k, where $k = 1$). Also from Table 2, we observe that the document-by-document method is more effective under comprehensiveness metrics, while the hour-by-hour method results are more effective under gain metrics. Examining the two different entity-focused features, entity importance and entity–entity interaction, from Table 2 we observe that both features exhibit very similar effectiveness under the harmonic mean metric. However, entity–entity interaction is more effective than entity importance, for both document-by-document and hour-by-hour, under normalised expected gain, although not when latency is taken into account. Further, entity importance is more effective than entity–entity interaction under comprehensiveness metrics for the document-by-document method.

We now examine results from Table 3, which presents evaluation scores from the automatic matching of non-pooled updates. Table 3 includes results for our 2 baseline run submissions, *uogTrdSqCR5* and *uogTrhSqCR6*, which were un-pooled. The system effectiveness ordering (as shown in Table 2) of our submitted runs, under harmonic mean, has not altered using this method of evaluation, but we note the track average has increased (from 0.0385 to 0.0472) due to the inclusion of un-pooled runs. This increase has resulted in only run *uogTrdSqCR5* exhibiting a significant improvement over the track average, under the harmonic mean metric, with runs *uogTrhEQR2* and *uogTrhEEQR4* exhibiting p-values of 0.0594 and 0.0506 respectively. The performance of the baselines, ranking sentences by their cosine similarity to the query, exhibit similar effectiveness over document-by-document and hour-by-hour methods, and similar to results in Table 2, hour-by-hour offers better gain, and document-by-document offers better comprehensiveness. The baseline runs, *uogTrdSqCR5* and *uogTrhSqCR6*, are used as input to the entity-focused runs, *uogTrdEQR1*, *uogTrhEQR2*, *uogTrdEEQR3* and *uogTrhEEQR4*, i.e. the entity-focused runs are re-ranking the baseline set of sentences. Under the harmonic mean metric, the re-ranking has led to a decrease in effectiveness for the document-by-document method, and has had little effect under the hour-by-hour method. However, examining the gain and comprehensiveness metrics, under document-by-document and hour-by-hour methods, we find that the re-ranking from the entity-focused event summarisation has led to improvements in the recall-oriented metric (comprehensiveness), but a loss in the precision-oriented metric (gain).

From the results in Table 2 and Table 3, we conclude that using entities to derive event summarisation features can lead to effective summaries of events. Further, the two entity-focused features we investigated performed broadly the same, and in future work a combination of features may lead to improvements in effectiveness. Additionally, we found that processing the corpus in hourly batches results in more effective event summary sentence selection decisions, possibly due to more information being available. Finally, we found that ranking sentences by their cosine similarity to the query, and selecting 1 sentence per batch boundary, offers a reasonably effective baseline for Task 3 of the track.

4. DYNAMIC DOMAIN

The primary aim of our participation in the first year of the Dynamic Domain track is to research and investigate the adaptation of resource selection and document prioritisation techniques for integration into our Terrier IR platform. More specifically, we explore two methods to minimise the number of iterations of the retrieval-feedback cycle needed to identify sub-topics that are of interest to the user. First, we a investigate a resource selection strategy to reduce the time taken to correctly identify the domain of interest for each query. Second, we investigate strategies to prioritise documents from the selected resources. Finally, we also investigate increasing the user's potential rate of exploration by diversifying the potential sub-topic intents presented to that user during each iteration. We summarise these methods in more detail below:

Resource Selection: We view the Dynamic Domain task as a resource selection problem, where each domain is considered as a separate resource. To prioritise resources, we use an implementation of CORI [2] to score each domain with respect to the frequency of the query terms within the domain. Having ranked the domains using resource selection, we investigate domain prioritisation strategies select documents to show to the user.

Domain Prioritisation Strategies: The domain prioritisation strategies that we investigate in our participation each select documents from all four domain resources to present to the user. However, each strategy apportions a different level of confidence based on the CORI ranking, by presenting the user with a proportionately higher percentage of documents from the top ranked domain. The details of the domain selection strategies are as follows:

- **Interleaving**: The Interleaving strategy attributes the least amount of confidence in the CORI ranking and therefore selects documents from all four domains to present to the user in each iteration. To prioritise resources, two documents from the top ranked domain and one document from each of the other of the three domains are selected to present to the user each in each iteration.
- **Round Robin**: The Round Robin strategy selects five documents from each domain resource within each iteration and presents the user documents from one resource at a time. The Round Robin strategy attributes a moderate degree of confidence to the CORI ranking, since if the domain of interest appears deeper in the CORI ranking then the number of iterations needed to present users documents from that domain is increased.
- **Multi-Armed Bandit**: For the Multi-Armed Bandit approach, we deploy a greedy approximation, namely

RunID	nE[Gain]	nE[Lat. Gain]	Comp.	Lat. Comp.	HM(nE[LG],Lat. Comp.)
TREC average	0.0420	0.0251	0.4551	0.2943	0.0385
uogTrdEEQR3	0.0438	0.0291	0.5662†	0.3823†	0.0528†
uogTrdEQR1	0.0419	0.0291	**0.6107**†	**0.4336**†	0.0533†
uogTrhEEQR4	**0.0732**†	0.0378†	0.4980	0.2812	0.0646†
uogTrhEQR2	0.0685†	**0.0381**†	0.5093	0.2981	**0.0654**†

Table 2: Performance of our submitted runs for Task 3, Summarisation Only, using the relevant only corpus.

RunID	nE[Gain]	nE[Lat. Gain]	Comp.	Lat. Comp.	HM(nE[LG],Lat. Comp.)
TREC average	0.0595	0.0319	0.5627	0.3603	0.0472
uogTrdEEQR3	0.0418†	0.0277	0.6096	0.4072†	0.0505
uogTrdEQR1	0.0402†	0.0275	**0.6590**†	**0.4614**†	0.0508
uogTrdSqCR5	0.0721	0.0363	0.4761†	0.2534†	0.0617†
uogTrhSqCR6	**0.1176**†	**0.0466**	0.3249†	0.1232†	0.0631
uogTrhEEQR4	0.0714	0.0365	0.5342	0.2983	0.0632
uogTrhEQR2	0.0667	0.0368	0.5459	0.3166	**0.0639**

Table 3: Un-pooled evaluation results, for Task 3, Summarisation Only, using the relevant only corpus.

Epsilon Greedy [17]. For each retrieval iteration, the probability of a document d being selected from the highest ranked domain resource, D_1, is $p(d_{D_1}) = \epsilon$ and the probability of d being selected from a randomly chosen domain that was not ranked highest by CORI is $1 - \epsilon$. For this run, we initially set $\epsilon = 1$ and decrease the value of ϵ in steps of 0.2 every third iteration until $\epsilon = 0.2$, at this point the run adopts the Interleaving approach for the remaining iterations. This approach initially assigns a high degree of confidence to the CORI ranking. The system becomes less confident in the CORI ranking as the number of iterations required to discover a relevant document increases.

Intent Diversification: Additionally we also investigate whether we can increase the user's exploration rate by increasing the diversity of content of the top ranked documents. One method to do so is to use search result diversification. There are two main approaches for diversifying search results, namely *explicit* and *implicit* diversification. Explicit search results diversification has previously been shown to perform well within the Web search domain [7, 11, 12]. Therefore, to maximise the number sub-topics presented to the user, we apply explicit search results diversification to each individual domain, before combining the rankings using a resource selection approach. To do this, we first identify potential sub-topic intents within a domain via topic modelling over the text of the top 30 ranked results returned from that domain. Then, we apply our state-of-the-art xQuAD diversification framework [15] to maximise these sub-topic intents within the ranking shown to the user. Having diversified the domain rankings, documents are then selected from each of the diversified domains in turn using the Round Robin approach described above.

Runs and Results: To evaluate the approaches described above, we submitted five runs to the main task of the Dynamic Domain track. For these runs, we use Terrier v4.0 to index the CBOR collection, removing stopwords and applying Porter stemming. We selected the classical tf*idf as our retrieval model and we set our stopping condition as the first iteration where the system returns a relevant document. The runs *uogTrSI*, *uogTrRR*, *uogTrIL* and *uogTrEpsilonG* evaluate the implemented resource selection techniques, while *uogTrxQuADRR* evaluates the benefits of search result diversification for maximising the user's exploration rate.

	Submitted	MRR	uERR	CT@10	ACT@10
TREC Median	-	-	0.2683	0.0575	0.1286
uogTrSI	✔	0.3506	**0.2317**	0.0269	**0.1699**
uogTrRR	✔	0.2535	0.1705	0.0228	0.1346
uogTrIL	✔	0.1687	0.1602	0.0184	0.1107
uogTrEpsilonG	✔	0.2434	0.1663	0.0215	0.1277
uogTrxQuADRR	✔	0.4038	0.2256	**0.0272**	0.0850
manual_R.S.	✖	0.5113	0.3120	0.0330	0.2222

Table 4: Results of our runs in the Dynamic Domain main track, the TREC median and a manual resource selection run. Figures in bold represent our top performing submitted runs.

- **uogTrSI**: This run uses a single index of all four domains and serves as our baseline.
- **uogTrRR**: This run uses CORI to rank domain resources and selects documents from resources based on the Round Robin strategy.
- **uogTrIL**: This run uses CORI to rank domain resources and selects documents from resources based on the Interleaving strategy.
- **uogTrEpsilonG**: This run uses CORI to rank domain resources and selects documents from resources based on the Multi-Armed Bandit strategy.
- **uogTrxQuADRR**: This run enhances the uogTrRR run by applying xQuAD search results diversification to ranking of documents within a domain before selecting documents from resources based on the Round Robin strategy.

Table 4 presents the performance of our runs submitted to the main task, along with the TREC median and a manual resource selection run that only selects documents from the correct domain for the query (manual_R.S.). The table reports the official track measures, uERR, Cube Test [8] (CT@10) and Averaged Cube Test (ACT@10), along with the mean reciprocal rank of the first relevant document(MRR). Firstly, we note from Table 4 that the single index baseline (uogTrSI) achieves a low MRR score (0.3506). Moreover, for this run, the first relevant document appears between ranks

151-992 for 23% of the topis. We suspect that this is mainly due to the low levels of completeness in the collection, as recognised by the track organisers. We also note that while tf*idf may not have been the optimal weighting model to deploy, it results in best completeness at ranks 100 and 1000 compared to a variety of weighting models implemented in Terrier. Secondly, we note from Table 4 that the resource selection approaches, *uogTrRR*, *uogTrIL* and *uogTrEpsilonG*, do not result in performance improvements over the single index baseline. This appears mainly to be due to the fact that CORI expects the query terms to have a higher relative frequency in the domain of interest than other domains. However, as we can see from the manual resource selection (manual_R.S.) run, that only ranks documents from the correct domain, a system that selects the correct domain resource for each query achieves notable performance improvements. Lastly, we conclude that search result diversification can improve the user's exploration rate, since we see that our xQuAD based run with Round Robin resource selection, *uogTrxQuADRR*, achieves higher uERR and CT@10 scores than the same approach without applying diversification, *uogTrRR*. Moreover, *uogTrxQuADRR* achieves a higher CT@10 score than the single index baseline.

5. CONCLUSIONS

In TREC 2015, we participate in both the "live" and "batch" experiments of the Contextual Suggestion track, the Summarisation Only task (Task 3) of the Temporal Summarisation track and the main task of the Dynamic Domain track. In particular, for the Contextual Suggestion track we propose two novel venue suggestion approaches that leverage data from the Foursquare LBSN. Firstly we propose an approach based on factorisation machines that uses check-in statistics, venue categories and user contexts to make personalised suggestions. Our second approach deploys a linear combination of a learning to rank technique and contextual appropriateness prediction. Overall, our runs are competitive, with both approaches performing above the TREC median. In particular, the factorisation machines run is the best of the two. For the Temporal Summarisation track, we explore entity-focused models for the summarisation of evolving events. In particular, we investigate *entity importance* and *entity–entity interaction* features. Furthermore, we also investigate two distinct methods of processing the corpus for event summarisation, namely *document-by-document* and *hour-by-hour* batches. Overall, our runs are competitive, with all submitted runs performing above the track average. Moreover, we note that processing the corpus in hourly batches results in more effective event summary sentence selection, and we conclude that using entities to derive event summarisation features can lead to effective summaries of events. For the Dynamic Domain track, we experiment with resource selection and document prioritisation strategies to reduce the number of iterations of the retrieval-feedback cycle needed to identify sub-topics that are of interest to the user. Moreover, we show that search result diversification can be used to increase the number of sub-topic intents within the document ranking and maximise the user's potential exploration rate.

6. REFERENCES

[1] J. Allan. Topic Detection and Tracking. chapter Introduction to Topic Detection and Tracking, pages 1–16. Kluwer Academic Publishers, 2002.

[2] J. P. Callan, Z. Lu, and W. B. Croft. Searching Distributed Collections with Inference Networks. In *Proc. of Conference on Research and Development in Information Retrieval*, SIGIR '95, 1995.

[3] R. Deveaud, M. Albakour, C. Macdonald, I. Ounis, et al. On the Importance of Venue-Dependent Features for Learning to Rank Contextual Suggestions. In *Proc. of the Conference Information & Knowledge Management*, CIKM '14, 2014.

[4] Q. Guo, F. Diaz, and E. Yom-Tov. Updating Users about Time Critical Events. *Advances in Information Retrieval*, 7814, 2013.

[5] K. Hong, J. Conroy, B. Favre, A. Kulesza, H. Lin, and A. Nenkova. A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization. In *Proc. of the Conference on Language Resources and Evaluation*, LREC '14, 2014.

[6] Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, 2009.

[7] 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 the Text REtrieval Conference*, TREC '12, 2012.

[8] J. Luo, C. Wing, H. Yang, and M. Hearst. The Water Filling Model and the Cube Test: Multi-dimensional Evaluation for Professional Search. In *Proc. of the Conference Information & Knowledge Management*, CIKM '13, 2013.

[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. L. Santos, I. Ounis, and B. He. About Learning Models with Multiple Query-dependent Features. *ACM Trans. Inf. Syst.*, 31(3):11:1–11:39, Aug. 2013.

[11] R. McCreadie, M.-D. Albakour, S. Mackie, N. Limosopathan, C. Macdonald, I. Ounis, and B. T. Dinçer. University of Glasgow at TREC 2013: Experiments with Terrier in Contextual Suggestion, Temporal Summarisation and Web Tracks. In *Proc. of the Text REtrieval Conference*, TREC '13, 2013.

[12] R. McCreadie, R. Deveaud, M.-D. Albakour, S. Mackie, N. Limsopatham, C. Macdonald, I. Ounis, T. Thonet, , and B. T. Dinçer. University of Glasgow at TREC 2014: Experiments with Terrier in Contextual Suggestion, Temporal Summarisation and Web Tracks. In *Proc. of the Text REtrieval Conference*, TREC '14, 2014.

[13] A. Nenkova and K. McKeown. Automatic Summarization. *Foundations & Trends in Information Retrieval*, 5(2-3), 2011.

[14] S. Rendle. Factorization Machine with libFM. *ACM Transactions on Intelligent Systems and Technology*, 3(3):1–22, 2012.

[15] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting Query Reformulations for Web Search Result Diversification. In *Proc. of the International World Wide Web Conference*, WWW '10, 2010.

[16] Y. Shi, M. Larson, and A. Hanjalic. List-wise Learning to Rank with Matrix Factorization for Collaborative Filtering. In *Proc. of the Conference on Recommender Systems*, RecSys '10.

[17] R. S. Sutton and A. G. Barto. *Reinforcement Learning: an Introduction*. MIT press Cambridge, 1998.

[18] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Ranking, boosting, and model adaptation. Technical report, Microsoft Research, 2008.