Overview of the TREC 2016 Contextual Suggestion Track

Seyyed Hadi Hashemi[1] Charles L.A. Clarke[2] Jaap Kamps[1] Julia Kiseleva[1] Ellen M. Voorhees[3]
[1]University of Amsterdam, Amsterdam, The Netherlands
[2]University of Waterloo, Waterloo, Canada
[3]NIST, Gaithersburg MD, USA

ABSTRACT

The TREC Contextual Suggestion Track offers a personalized point of interest (POI) recommendation task, in which participants develop systems to give a ranked list of suggestions related to a profile and a context pair available in the tasks' requests provided by the track organizers. Previously, reusability of the contextual suggestion track suffered from using dynamic collections and a shallow pool depth. The main innovations at TREC 2016 are the following. First, the TREC CS web corpus, consisting of a web crawl of the TREC contextual suggestion collection, was made available. The rich textual descriptions of the web pages makes far more information available for each candidate POI in the collection. Second, we released endorsements (end user tags) of the attractions as given by NIST assessors, potentially matching the endorsements of POIs in another city as given by the person issuing the request as part of her profile. Third, a multi-depth pooling approach extending beyond the shallow top 5 pool was used. The multi-depth pooling approach has created a test collection that provides more reliable evaluation results in ranks deeper than the traditional pool cut-off.

1. INTRODUCTION

The TREC Contextual Suggestion Track ran for the fifth and last year as an independent track in 2016 [4–7]. The track has the primary goal of providing reusable test collection for evaluation of point-of-interest (POI) recommendation systems. The test collection is open to anyone who is willing to do research in contextual suggestion problem.

The contextual suggestion track assumes a traveller in a specific context (e.g., a city and trip type) seeking things to do that reflects their own interests, which is supposed to be inferred from their interests in the given context and a visited city (seed cities in the track). Given a user's contexts and profile including a POI list, their tags/endorsements, and ratings from the seed cities, participants make recommendations for attractions in a new context (including the target city as the location).

For example, imagine a group of information retrieval researchers with a November evening to spend in beautiful Gaithersburg, Maryland. A contextual suggestion system might recommend a beer at the Dogfish Head Alehouse[1], dinner at the Flaming Pit[2], or even a trip into Washington on the metro to see the National Mall[3].

If you are familiar with the track, which has been operated since 2012, the main changes in this year is listed as follows:

1. The track provides a fixed TREC Contextual Suggestion Web corpus as an additional data to overcome the dynamic nature of the open web.
2. The track provides endorsements (i.e., tags) of venues.
3. The track was split into two phases:
   (a) Phase 1 experiment, which is a collection based task similar to the TREC 2015 Contextual Suggestion Track's Live Experiment. The main change is that the track does not require participants set up and register a live server. However, the track distributes a set of profiles and contexts and collect responses in a batch wise fashion, as was used in the track until 2014.
   (b) Phase 2 experiment, which is a reranking task similar to the TREC 2015 Contextual Suggestion Track's Batch Experiment.
4. The track used a multilayer pooling approach that aimed creating a reusable test collection, which was very challenging in previous years of the track [10, 12].

The rest of this paper is organized in the following way. Next, in §2, we will detail the track's tasks. This is followed by a discussion of the resulting test collection in §3 and the pooling method in §4. Then, §5 details the evaluation results of all submissions and teams. We conclude the paper in §6.

2. TASK OVERVIEW

This section will discuss the tasks of the TREC 2016 contextual suggestion track.

The track followed the setup of 2015 with two distinct phases. In both phase 1 and phase 2 tasks, participants were asked to develop a system that is able to make suggestions for a specific person based on their given profile and context. As input of the task, the track organizers provide a set of profiles, a set of contexts and a set of example suggestions (URLs of pages corresponding to POIs in a given context). Each profile corresponded to a single user's preferences in example suggestions of another context or city, their gender and age, and each context includes information about the target city (i.e., the target location), a trip type, a

[1]www.dogfishalehouse.com
[2]www.flamingpitrestaurant.com

Table 1: TREC Contextual Suggestion track collection example.

Attraction ID	City ID	URL	Title
TRECCS-00000005-418	418	http://www.greatfallsmt.net/people_offices/park_rec/gibson.php	"Gibson Park"
TRECCS-00000006-418	418	http://www.mackenzieriverpizza.com	"MacKenzie River Pizza Co"
TRECCS-00000007-418	418	http://www.bostons.com	"Bostons Restaurant Sports Bar"
TRECCS-00000008-418	418	http://pink.victoriassecret.com	"Victorias Secret PINK"

trip duration, a type of group the person is travelling with, and a season the trip will occur in.

Profiles correspond to the stated preferences of real individuals, who either recruited through crowdsourcing or recruited editorial judges. These assessors first judged example attractions in seed locations, later returning to judge suggestions proposed by the phase 1 participants for various contexts. Both for the profile (i.e., seed pages) and for the suggested recommendations, assessors were able to choose the context or city for which recommendations were judged.

As output of the phase 1 task, for each context/profile pair, participants were required to return a ranked list of 50 suggestions. Each suggestion was expected to be relevant to the given profile and the context. As output of the phase 2 task, participants were expected to rerank the given suggestion candidates with respect to the user's profile and context and return them as the phase 2 response. To be precise:

Phase 1 Experiments The phase 1 experiment is a collection based task, in which participants are asked to develop a contextual suggestion system that is able to make suggestion for a particular person in a specific context. In particular, for each given request (including profile and context), participants has to retrieve 50 suggestions from the TREC contextual suggestion collection as a response.

Phase 2 Experiments The phase 2 experiment is a reranking task, in which a suggestion candidates set is provided for each request. In fact, all the suggestion candidates available in phase 2 requests were made by participants in phase 1. Therefore, we have all the judgments of the suggestions available in the suggestion candidates, which facilitates the reuse of the contextual suggestion test collection.

The track continues to use a collection of URLs corresponding to POIs in each context that was released in 2015, see the examples in Table 1. For the future studies on the contextual suggestion problem using the TREC contextual suggestion track qrels, due to the dynamic nature of the collection, we strongly recommend to use the TREC Contextual Suggestion Web corpus, which will be introduced in Section 3.2.

3. TEST COLLECTION

This section discusses the resulting test collection.

TREC 2016 contextual suggestion test collection consists of a corpus (including TREC contextual suggestion collection and the web corpus), a set of requests, and relevance judgments. In addition we have also released suggestions' endorsements.

[3]www.nps.gov/nacc

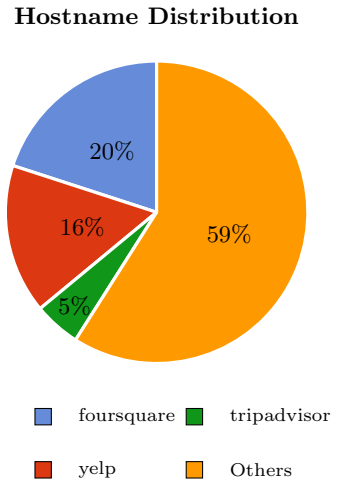

Figure 1: Most popular domains in the TREC Contextual Suggestion Web Corpus.

3.1 TREC CS Collection

The TREC Contetxual Suggestion collection was collected by asking participants as volunteers to retrieve suggestion candidates related to each city from the open web in a pre-task phase. This collection was created in TREC 2015 contextual suggestion track. The collection consists of a set of attractions. For each attraction there are:

1. An attraction ID, which contains three parts separated by dashes (-)
   (a) The string 'TRECCS'
   (b) An 8 digit number
   (c) A three digit number corresponding to that attraction's city ID
2. A city ID which indicates which city this attraction is in
3. A URL with more information about the attraction
4. A title

An example of the TREC Contextual Suggestion collection is given in Table 1.

3.2 TREC CS Web Corpus

In addition to the TREC contextual suggestion collection, which is available since 2015, we released TREC contextual suggestion web corpus. The TREC CS web corpus is a web crawl of the suggestions' URLs available at the TREC contextual suggestion collection. In this crawl, we have managed to fetch 77.39 % of the whole TREC Contextual Suggestion collection, which is 956,437 web pages out of 1,235,844 URLs.

```
{"id":743,
"body": {
    "group": "Friends",
    "season":"Summer",
    "trip_type":"Holiday",
    "duration":"Weekend trip",
    "location":{
        "state":"TX",
        "id":306,
        "name":"Waco",
        "lat":31.54933,
        "lng":-97.14667},
    "person": {
                "gender": "Male",
                "age": 28,
                "id": 15012,
                "preferences":[
                    {
                    "rating":4,
                    "documentId":"TRECCS-00211395-161",
                    "tags":[
                         "Beer",
                         "Culture",
                         "Cocktails",
                         "Restaurants",
                         "Food",
                         "pub-hopping",
                         "cocktails",
                         "bar-hopping"
                         ]},
                    ...
                    ]
                    }},
"candidates":[
    {"documentId":"TRECCS-00267253-306",
    "tags":[
        "Beer",
        "Cocktails",
        "Family Friendly",
        "Restaurants",
        "Food"
        ]},
    {"documentId":"TRECCS-00294259-306",
    "tags":[
    "Tourism",
    "Bar-hopping",
    "Restaurants",
    "Entertainment",
    "Live Music"
    ]},
     ...
    ]
    }
```

Example 1: TREC Contextual Suggestion Track phase 2 request example in JSON format

This crawl includes web pages from different domains like yelp, tripadvisor and foursquare. Yelp was the most difficult domain to crawl, and we managed to crawl about 153K out of 220K yelp web pages available in the TREC contextual suggestion collection. Figure 1 indicates percentage of available POIs from the most popular tourist attraction domains in the TREC Contextual Suggestion Web corpus. As it is shown in this figure, Foursquare, Yelp and Tripadvisor are the most popular domains in the TREC Contextual Suggestion Web corpus.

The TREC Contextual Suggestion Web Corpus includes attraction web pages of 272 different North American cities. In this corpus, there are 3,516.31 tourist attraction web pages in average per city. The corpus is in a WARC (Web ARChive) format. In order to have access to the data designated as the TREC CS Web Corpus, organizations must first fill in a data release Organizational Application Form. Then, the signed form must be scanned and sent by email to data@list.uva.nl. On receipt of the form, participants will be sent information on how to download the corpus.

3.3 Requests

In both phase 1 and phase 2 experiments, each request contains information about assessors' preferences as profiles and their chosen context. Moreover, phase 2 requests contains suggestion candidates related to each profile and context pair. Each profile consists of a list of attractions the assessor has previously rated, their gender and their age. For each attraction the profile will include:

1. A rating:
   (a) 4: Strongly interested
   (b) 3: Interested
   (c) 2: Neither interested or uninterested
   (d) 1: Uninterested
   (e) 0: Strongly uninterested
   (f) -1: Not loaded or no rating given
2. Tags/endorsements if it is applicable.

Each context consists of a city name which represents which city the trip will occur in and several pieces of data about the trip. The context is as follows:

1. A city the trip will occur in (e.g., Seattle)
2. A trip type (e.g., Business)
3. A trip duration (e.g., Weekend trip)
4. A type of group the person is travelling with (e.g., Travelling with a group of friends as "Friends")
5. A season the trip will occur in (e.g., Summer)

An example of the TREC Contextual Suggestion phase 2 request is shown in Example 1. The track organizers provide 438 input requests in total, in which requests having identifiers from 700 to 922 are used for the official experiments in TREC 2016 contextual suggestion track. In particular, TREC 2016 Phase 1 test collection consists of judgments of 61 requests, and TREC 2016 Phase 2 test collection includes all the phase 1 requests except requests having 707, 912 and 922 as identifiers, hence 58 requests in total. The difference is a result of some additional judged requests coming available after the release of the phase 2 requests. Some examples of official phase 1 requests' context and profile statistics are shown in Figure 2.

In building profiles for the TREC 2016 official requests (request IDs >= 700), two seed cities were chosen (Seattle and Detroit). Each seed city had 30 POIs to be judged as user profiles. Users could choose which seed city to judge. If they just rate POIs of one of the cities, their profiles have 30 rated POIs. If they rate both of the seed cities' POIs, their profiles have 60 rated POIs. For example, in Phase 2 official requests, there are 39 requests having 30 judged example suggestions and 19 requests having 60 judged example suggestions in their profiles.

In phase 2 requests, due to the use of multi-depth pooling, which will be detailed in Section 4, the size of provided suggestion candidates is varied per request. Specifically, average number of suggestion candidates over the 58 phase 2 requests is 96.53, maximum number of suggestion candidates is 119 and minimum number of suggestion candidates is 79.

The rest of the requests, which were collected in TREC 2015, were used as train set of the TREC 2016 contextual suggestion track, as the qrels of those requests were available since TREC 2015. The TREC 2016 identifiers of those requests are same as the one used in TREC 2015, which facilitates evaluation of these requests based on the TREC 2015 contextual suggestion test collection. However, we have created a new pool and new sets of suggestions as suggestion candidates using the multi-depth pooling approach, which will be discussed in Section 4. Therefore, suggestion candidates of those requests available in TREC 2015 are different from the ones in TREC 2016. In fact, TREC 2015 batch requests contain a set of suggestion candidates with a very high probability of being relevant to the request. To make it a more realistic and challenging problem, we have injected more noise into the original batch requests of TREC 2015, hence the sets of candidates for the 2015 requests included this year differs from those of last year.

There are further requests that are based on requests made during the TREC 2015 live tasks. There were left out of the TREC 2015 data, privileging only a single request per crowdsourced assessor, but judgement are available to be used. As these requests were not as deeply pooled as the official TREC 2016 requests, they are excluded again from the official test collection in 2016, but may be released separately at a later date.

3.4 Relevance Judgments

Relevance judgments were collected through crowdsourcing and by the help of a group of graduate students. They were asked to rate suggestions in a same scale that presented in Section 3.3.

However, in the qrels, we have shifted the raw assessors' 5 point scale judgments with -2, making the judgments in the range -3 to 2, and making a score of 1.0 or higher correspond to a "interested" or "strongly interested" judgment. Therefore, the trec_eval can be used to evaluate contextual suggestion runs based on all the common IR measures, included graded measures like NDCG.

3.5 Suggestions Endorsements

In addition to the relevance judgments based on the ratings, we also asked the assessors to endorse the suggestions

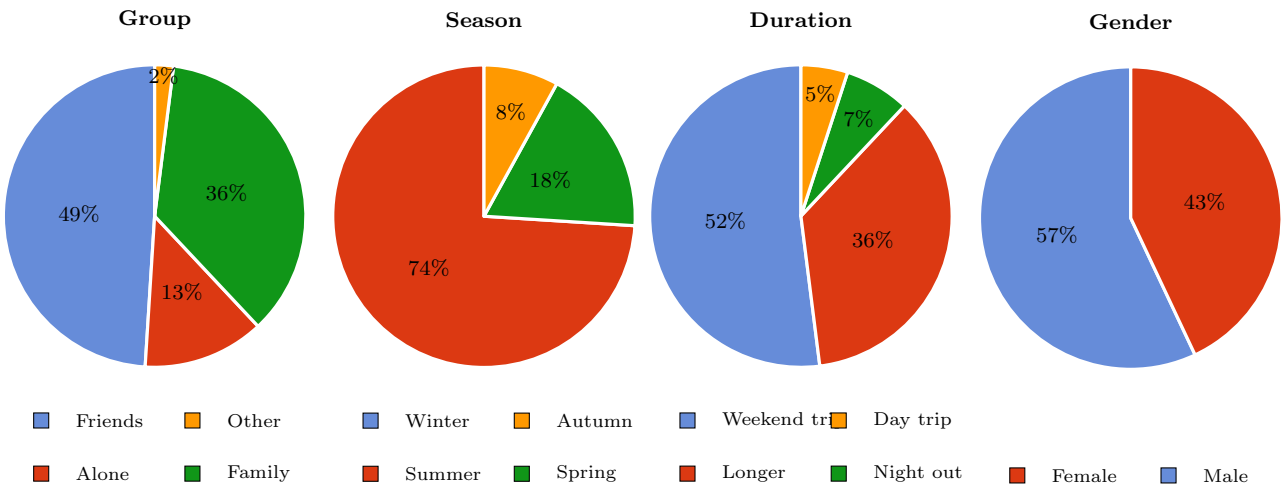

Figure 2: Example of official phase 1 requests' contexts and profiles statistics.

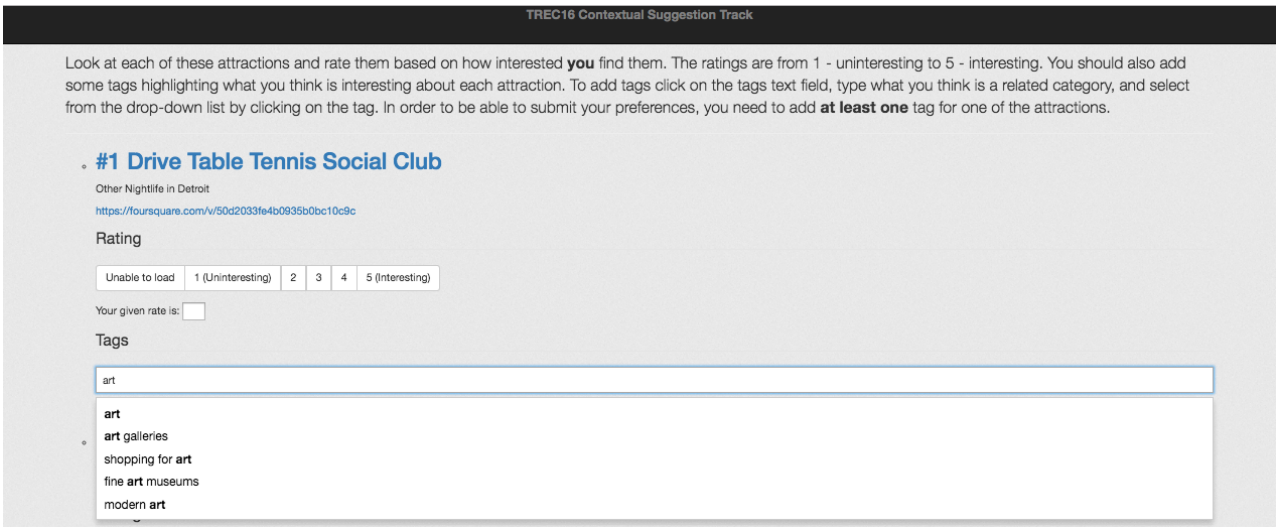

Figure 3: An example of how assessors give rating and tags/endorsements to the suggestions.

using the tag field, which is shown in Figure 3.

In practice, endorsement was not an easy task for them, and they were not willing to give tags to all the given suggestions. Therefore, NIST assessors endorsed all the pooled suggestions, and we include those tags/endorsements to both profiles and suggestion candidates of the phase 2 requests.

4. POOLING APPROACH

This section discusses the pooling approach used at TREC 2016.

Previously, TREC contextual suggestion organizers used the traditional pooling approach and pooled all the top-N suggestions of the submissions, in which N is a pool cut-off. They created a pool using 5 as the pool cut-off. According to the studies done on the reusability of the TREC contextual suggestion test collection [9–12], reusability of the test collection suffered a lot from the personalization effects and respectively the shallow pool cut-off. To address this issue, we experimented with a "multi-depth" pooling approach.

4.1 Multi-Depth Pooling

In the multi-depth pooling approach, in addition to the pool cut-off (hard pool cut-off), they defined two others pool cut-offs, namely, soft pool cut-off and very soft pool cut-off. In the multi-depth pooling approach, they have pooled the following suggestions:

1. All the suggestions/documents ranked higher than the hard pool cut-off by any of the submissions is pooled. This would guarantee an stable measures up to the traditional pool cut-off.

2. In addition, if a suggestion/document ranked higher than the soft pool cut-off by at least one submission, and also ranked higher than the very soft pool cut-off by at least one run from another participated team, the suggestion is pooled. This would have effects on having more stable measures deeper than the traditional hard pool cut-off in the ranking.

Following last years of the TREC contextual suggestion track, we have used 5 as the hard pool cut-off. In addition, taking

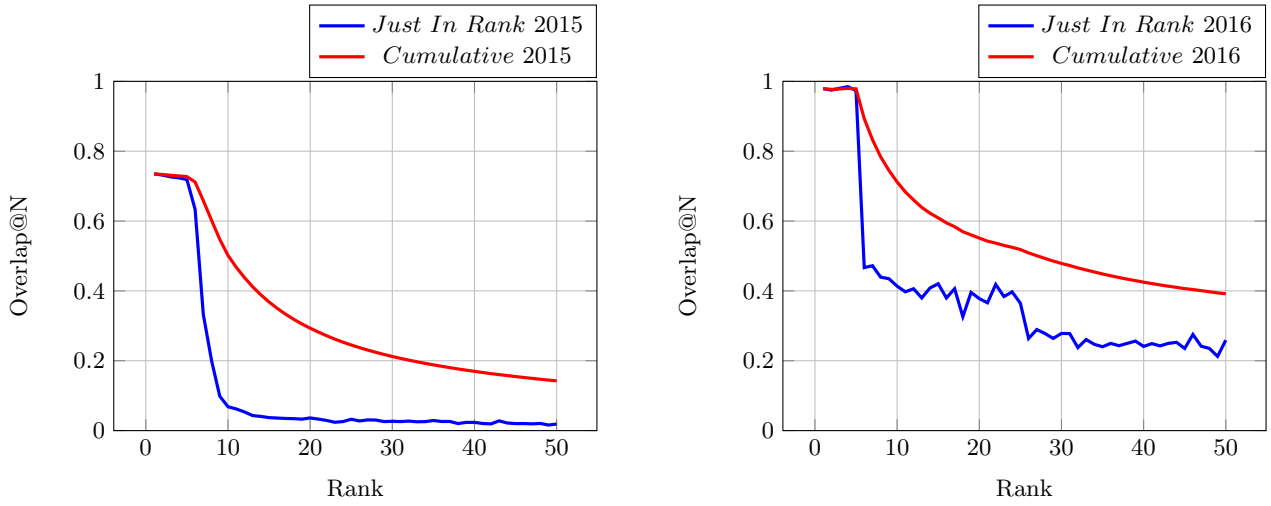

Figure 4: Cumulative and just-in-rank Overlap@N in TREC 2015 and 2016 contextual suggestion test collections.

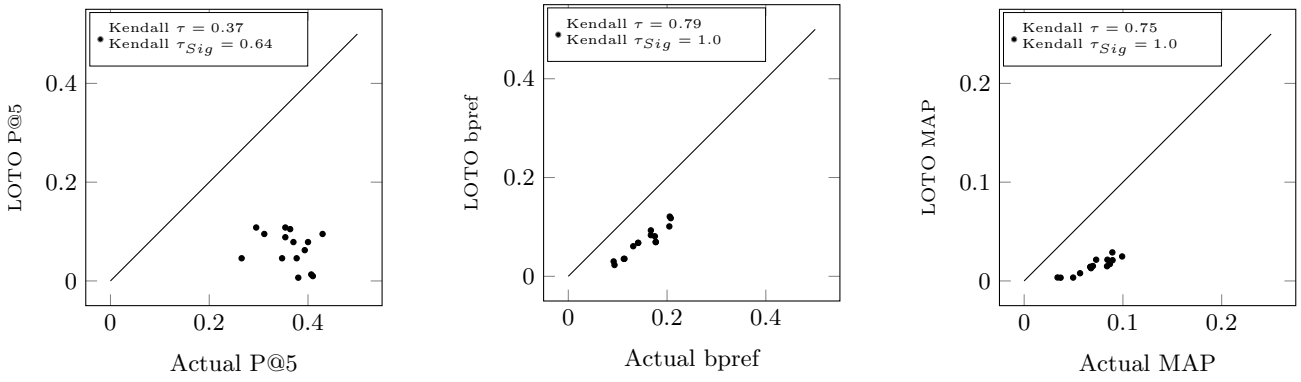

Figure 5: Leave One Team Out (LOTO) reusability test of the contextual suggestion test collection created based on multi-depth pooling.

into account the effort needed to create the test collection, we have set 25 as the soft pool cut-off and 50 as the very soft pool cut-off as this leads to a pool size of about 100 suggestions per request.

The proposed pooling approach would give us more stable evaluation results over deeper ranks than the traditional pool cut-off. The traditional pooling approach with 5 as the pool cut-off would cost 3,377 judgments for the 61 official phase 1 requests. Interestingly, the above multi-depth pooling approach spend even less effort than pooling top-10 documents/suggestions provided by the submissions. Specifically, for the official qrels of the TREC 2016 contextual suggestion, we have collected 5,898 judgments using multi-depth pooling approach, in which we have got 5,782 official judgments after filtering some noises. If we had used the traditional pooling approach with 10 as the pool cut-off, we would have collected 6,206 judgments.

4.2 Fraction of Judged Documents

In multi-depth pooling, we have pooled deeper and expected a larger fraction of judged documents after the pool cut-off. Figure 4 shows a comparison of the cumulative over-lap@N [10] in TREC 2015 and 2016 Contextual Suggestion tracks. As it is shown in Figure 4, the fraction of judged documents is gently decreases after the hard pool cut-off (i.e., 5) in TREC 2016 contextual suggestion test collection. However, in TREC 2015 contextual suggestion track, fraction of judged documents dropped dramatically after the pool cut-off (i.e., 5). We have also plotted just-in-rank overlap@N in Figure 4, in which we just consider fraction of judged and unjudged documents at rank N and calculate the overlap. This figure indicates that the multi-depth pooling is effective in minimizing the fraction of unjudged documents in ranks deeper than the pool cut-off. The larger fraction of judged documents in TREC 2016 helps us to have a more stable evaluation over ranks deeper than the traditional pool cut-off.

4.3 Reusability

As shown in Figure 4, the fraction of judged documents has improved in ranks deeper than the hard pool cut-off using multi-depth pooling. However, effects of this improvement on the reusability of the test collection are not a priori clear.

Figure 5 demonstrates reusability of the TREC 2016 test collection based on Leave-One-Team-Out (i.e., LOTO) [3] test. According to Figure 5, the TREC 2016 contextual suggestion test collection should be used with some care based on P@5 metric. The official runs are completely judged up to rank 5, by design of the pooling approach, but post-submission experiments not contributing to the pool of judged documents risk being underrated. We have observed a similar system ranking correlation based on NDCG@5 metric having Kendall's $\tau = 0.43$.

There is also good news: the test collection appears to be reusable when considering the more stable evaluation measures for incomplete test collections. Specifically, the test collection has got perfect system ranking correlation between official TREC system ranking and the LOTO system ranking based on the Kendall's τ using statistical significant inversions using MAP and bpref metrics. In this test, 54% of the pairwise comparisons are significant based on MAP and we have had 64% significant differences based on bpref.

5. EVALUATION RESULTS

In this section, we first list our official evaluation measures. Then, we detail the evaluation results of phase 1 and 2 experiments.

5.1 Evaluation Measures

Three measures are used to rank both phase 1 and phase 2 runs. Our main measure is NDCG@5; in addition, P@5 and MRR are also used as two other metrics have been used since 2012 in TREC contextual suggestion track. As early rank cut-off measures are notably unstable, we also include measures taking more of the ranking into account, such as P@10, NDCG, MAP, Rprec and bpref, also profiting from the deeper pooling approach of this year.

The official results for the phase 1 task are shown in Table 2. The best phase 1 runs from top-5 teams out of 8 participated teams in phase 1 will be detailed in Section 5.2. Table 3 shows the official results for the phase 2 task. The best phase 2 runs from top-5 teams out of 13 participated teams in phase 2 will be summarized in Section 5.3.

5.2 Best Performing Phase 1 Submissions

The five best performing teams in the phase 1 evaluation are the following:

5.2.1 USI

USI [1]'s best performing phase 1 run is "USI2", in which they crawled Foursquare for virtually 600K venues. Using the crawled data, they created positive and negative category profiles consisting of all categories a user liked/disliked as well as their corresponding normalized frequencies. The initial category profiles are then used to measure the similarity between a new venue and a particular user. They created the initial ranking and picked the top 10 venues for each user to gather extra information about them. For each user they also created positive and negative frequency-based venue taste keyword profiles. For the new set of venues, they extracted venue taste keywords and measured the similarity between the venues and a particular user. They reranked the top 10 venues for each user in the initial ranking using a linear combination of the venue category and taste keyword scores

5.2.2 IAPLab

Nanjing University's IAP Lab did not provide a description of their approach by the time of writing, nor submitted a participants' paper to the TREC Notebook or TREC Proceedings. Therefore, we cannot provide a further description of their approach in the overview paper, apart from noting that their system did well for the phase 1 task.

5.2.3 ADAPT_TCD

ADAPT_TCD [2] proposed an ontology-based approach, using an ontology that was constructed using the Foursquare Category Hierarchy. The three models, each based upon this ontology, are: User Model, Document Model and Rule Model. For the User Model they build two models, one for each phase of the task, based upon the attractions that were rated in the user's profile. In the first phase they use only the positively rated attractions from each user. In the second phase they use both positive and negatively rated attractions to build the user model. The Document Model enriches documents with extra metadata (tags) from Foursquare and categories (concepts) from the ontology are attached to each document. The Rule model is used to tune the score for each candidate suggestion based upon the context of the trip and how it aligns with the rules in the model.

Their best performing run is "ADAPT_TCD_r1" in which, they build the user positive model based on the positively rated attractions in the user's profile. For each of these attractions, they create an index of all the classes, based on Foursquare data, that these attractions are an instance of, along with the tag set that was found on that attraction's page on Foursquare. They then compute the count per class and then the percentage of each class in the positive model. For a given place p that a user is travelling to, they select the documents that match the classes in the positive model. They eliminate the documents that belong to a class that violates at least one rule in the rule model. They retain the class percentage breakdown from the user model and map these percentages to 50 and represented this as a number, x, for each class. Following this, they select the top x attractions of this class from the retrieved documents after ranking them based on the features that have been collected in the Document Model from Foursquare, which are: the average users' rating, the users' rating count, the users' reviews count and the tag similarity measure between a document's tag set and the class tag set. After they select the required number of documents for all classes in the user model, they start to rank the documents based on the first three features mentioned before and return the final ranked list. If the number of attractions belonging to a specific class, in a specific city, do not meet the required number, they compensate for the shortfall by getting more attractions from the highest ranked class/classes in the user model.

5.2.4 FUM-IRLAB

FUM-IRLAB [15] followed two main approaches for finding suitable attractions for a given user: a content-based approach and a category-based approach.

In the content-based approach, all Web pages related to attractions are modeled as vectors of real numbers using word embedding and document embedding techniques. Then, similarities between attractions in the profile of a given user and new attractions are calculated using methods for finding similarities between vectors.

Table 2: Official TREC 2016 Contextual Suggestion Track's *phase 1 submissions* evaluated over 61 requests.

Rank	RunID	**NDCG@5**	P@5	MRR	NDCG	MAP	bpref	P@10	Rprec
1	USI2	*0.2826*	*0.4295*	0.6150	0.2083	0.0868	0.1772	0.3148	0.1619
2	IAPLab1	0.2789	0.3770	*0.6245*	0.2000	0.0729	0.1672	0.2721	0.1458
3	ADAPT_TCD_r1	0.2643	0.4066	0.5777	*0.2333*	*0.0992*	0.2046	*0.3246*	0.1886
4	FUM-IRLAB_3	0.2601	0.3803	0.5824	0.1494	0.0566	0.1124	0.2623	0.1133
5	FUM-IRLAB_1	0.2596	0.4000	0.5501	0.1928	0.0696	0.1672	0.2721	0.1498
6	ADAPT_TCD_r2	0.2595	0.4098	0.5512	0.2088	0.0895	0.1753	0.3230	0.1770
7	USI1	0.2578	0.3934	0.6139	0.2030	0.0839	0.1769	0.3148	0.1578
8	FUM-IRLAB_2	0.2544	0.3705	0.5945	0.1719	0.0677	0.1315	0.2885	0.1356
9	ExPoSe_response_tags	0.2461	0.3639	0.5206	0.1398	0.0496	0.1138	0.2033	0.0926
10	ExPoSe_response_all	0.2445	0.3541	0.5128	0.1735	0.0672	0.1413	0.2393	0.1282
11	ExPoSe_response_content	0.2443	0.3541	0.5114	0.1731	0.0669	0.1416	0.2393	0.1278
12	bupt_runA	0.2395	0.3475	0.5366	0.2255	0.0843	*0.2075*	0.2689	*0.1899*
13	UAmsterdam1	0.2026	0.2951	0.4387	0.1169	0.0369	0.0936	0.1754	0.0803
14	Laval_run1	0.1932	0.3115	0.4391	0.2209	0.0893	0.2054	0.2770	0.1936
15	UAmsterdam2	0.1641	0.2656	0.4095	0.1046	0.0338	0.0918	0.1607	0.0788

Table 3: Official TREC 2016 Contextual Suggestion Track's *phase 2 submissions* evaluated over 58 requests (excluding 707, 912, 922).

Rank	RunID	**NDCG@5**	P@5	MRR	NDCG	MAP	bpref	P@10	Rprec
1	DUTH_rocchio	*0.3306*	0.4724	0.6801	*0.6835*	0.4497	0.4704	0.4552	0.4245
2	Laval_batch_3	0.3281	*0.5069*	0.6501	0.6770	0.4536	0.4666	0.4500	0.4168
3	USI5	0.3265	*0.5069*	0.6796	0.6804	0.4590	0.4507	*0.4603*	0.4177
4	DUTH_bcf	0.3259	0.4724	0.5971	0.6829	*0.4606*	*0.4845*	0.4431	*0.4312*
5	USI4	0.3234	0.4828	*0.6854*	0.6813	0.4576	0.4494	0.4552	0.4229
6	Laval_batch_2	0.3118	0.4345	0.6287	0.6746	0.4378	0.4721	0.4207	0.4158
7	DUTH_knn	0.3116	0.4345	0.6131	0.6763	0.4456	0.4825	0.4448	0.4189
8	bupt_pris_2016_cs.2_.4_max	0.2936	0.4483	0.6255	0.6625	0.4318	0.4476	0.3983	0.3956
9	Laval_batch_1	0.2889	0.4276	0.6372	0.6680	0.4397	0.4409	0.4310	0.4246
10	UAmsterdamDL	0.2824	0.4448	0.5924	0.6544	0.4168	0.4452	0.4310	0.3881
11	bupt_pris_2016_cs.4_.2_max	0.2761	0.4241	0.5937	0.6602	0.4308	0.4465	0.4155	0.4031
12	DPLAB_IITBHU_iitbhu01	0.2757	0.4138	0.6298	0.6594	0.4269	0.4461	0.4034	0.4042
13	uogTrCs	0.2756	0.4207	0.5886	0.6585	0.4253	0.4500	0.3983	0.4005
14	UAmsterdamCB	0.2730	0.4069	0.5631	0.6499	0.4076	0.4337	0.4000	0.3780
15	ADAPT_TCD_br1	0.2720	0.4241	0.5472	0.6570	0.4357	0.4350	0.4103	0.4065
16	ADAPT_TCD_br2	0.2720	0.4241	0.5472	0.6570	0.4357	0.4328	0.4103	0.4068
17	SCIAICLTeam_CasualChocolate	0.2650	0.3828	0.5853	0.6574	0.4213	0.4278	0.3931	0.3885
18	IAPLab2	0.2615	0.4034	0.5635	0.6524	0.4140	0.4547	0.3828	0.3934
19	ADAPT_TCD_br3	0.2612	0.3931	0.5996	0.6585	0.4342	0.4366	0.4034	0.4090
20	uogTrCsContext	0.2582	0.3828	0.5475	0.6566	0.4265	0.4454	0.4052	0.4058
21	SCIAICLTeam_SassyStrawberry	0.2543	0.3690	0.5931	0.6556	0.4189	0.4275	0.3810	0.3863
22	bupt_pris_2016_cs.3_.3_avg	0.2471	0.3793	0.6014	0.6505	0.4186	0.4396	0.3862	0.3879
23	USI3	0.2470	0.4103	0.6231	0.6596	0.4425	0.4471	0.4259	0.4151
24	ExPoSe_SWLM	0.2375	0.3448	0.5285	0.6526	0.4125	0.4467	0.3845	0.3979
25	DPLAB_IITBHU_iitbhu04	0.2325	0.3310	0.5367	0.6507	0.4145	0.4363	0.3741	0.3933
26	FUM-IRLAB_phase2_2	0.2318	0.3655	0.5191	0.6376	0.3985	0.4357	0.3759	0.3732
27	FUM-IRLAB_phase2_1	0.2298	0.3517	0.5335	0.6378	0.3974	0.4344	0.3776	0.3696
28	SCIAICLTeam_VerbatimVanilla	0.2119	0.3310	0.5371	0.6463	0.4099	0.4477	0.3707	0.3916
29	DPLAB_IITBHU_iitbhu05	0.2106	0.3034	0.4921	0.6347	0.3923	0.4207	0.3362	0.3638
30	CityUHKGeng_1st_subminssion	0.1662	0.2414	0.3357	0.3882	0.2119	0.3312	0.2483	0.2157

In the category-based method, a subset of attractions is modeled as a vector of categories. These categories are extracted from the category information of the related Yelp, TripAdvisor, or Foursquare pages of the attractions. In addition, a user profile is modeled as a vector of categories, where these are categories extracted based on a mapping from the tags provided in the user's profile and the categories extracted for the attractions. Finally, similarities between attractions and user profiles are calculated based on similarities between these vectors. They submitted three methods of combining these two approaches to this track as three different runs.

Their best performing run is "FUM-IRLAB_3", in which the document-embedding vectors and the similarities between them are employed to produce a list of the most similar attractions to each attraction in the user profile. They found that despite a lot of very related results, this list contains a couple of completely unrelated pages. Hence, they decided to filter the result set for having a more precise list of attractions. They made an intersection between these lists with the attractions provided by category-based approach, making them more precise in the cost of decreasing recall. For each liked attraction in the user profile, they created a list of similar attractions, and then they iteratively selected two top attractions from each list and merged them to the final result set. They continue their iterations until they find 50 results from these lists.

5.2.5 ExPoSe

ExPoSe [8] focused on one of the key steps of contextual suggestion methods is estimating a proper model for representing different objects in the data like users and attractions. They used the Significant Words Language Models (SWLM) as an effective method for estimating models representing significant features of sets of attractions as user profiles and sets of users as group profile. The SWLM model outperformed the standard language model, and is robust against negative examples.

For phase 1, the tag based run "ExPoSe_response_tags" obtained a better score than the content-based, and the combined run—although the differences between the runs were small.

5.3 Best Performing Phase 2 Submissions

The five best performing teams in the phase 2 evaluation are the following:

5.3.1 DUTH

DUTH [14] have further developed and built upon the two methods they first presented in Contextual Suggestion 2013, which they have fine-tuned using TREC 2015 data. They address the task by individually using two classification methods, namely, a weighted k-NN classifier and a modified Rocchio classifier. Also, as a third method, they explore the use of election systems, namely Borda Count, as a means of fusing the results of the two aforementioned classifiers.

Their best performing run is "DUTH_rocchio", which is based on a Rocchio-like classifier. Using a user's rated venues as training examples, they build a custom query for the user using a modified Rocchio relevance feedback method. Specifically, they build a centroid per rating and combine/add those using their corresponding ratings as contributing factors, offset by 2 so as ratings 0 and 1 provide negative feedback with -2 and -1 weights respectively. Rating 2 is eliminated as neutral.

5.3.2 LavalLakehead

LavalLakehead [16] formulate a customized query according to user profile to retrieve the 100 initial attractions. Then these 100 candidates are ranked by two independent ranking models who cover global trend of interests and contextual individual preference respectively. The first model is a pre-trained regressor on 2015 TREC data thus it can prioritize popular places and categories loved by all users (E.g. Museums and National Parks). The second model introduces word embedding to captures individual user preference. Both user profiles and candidate places are represented as word vectors in a same Euclidean space. So that a similarity score between user and attraction can be calculated by measuring their vector distance. In the end, a final ranking is given by summing up the two models' scores, and "Laval_batch_3" is a result of the combination of the two above models.

5.3.3 USI

USI [1]'s best performing phase 2 run is "USI5", in which they computed a set of multimodal scores from multiple locationbased social networks (LBSNs) and combined them with a score that predicts the level of appropriateness of a venue to a given user context. Briefly, the scores are calculated as follows: positive and negative reviews are used to create user profiles to train a classifier which then predicts how much a particular user will like a new venue. Moreover, the frequency-based scores are calculated based on the venue categories and taste keywords. As for the prediction of appropriateness, they created two datasets using crowdsourcing and trained a classifier with the features they extracted from the datasets. A linear combination of all the scores produced the final ranking of the candidate suggestions.

5.3.4 bupt_pris_2016

BUPT [18] collected data by crawling from the Yelp API and Foursquare API. With attractions marked with rating and tags in the preference list, they calculated users' average rating for each tag. For tags without a rating of the user in the profile, that is, the missing ratings, they filled them by Collaborative Filtering. Next, they got the users' rating for an attraction with either a mean function or a max function. By ranking the ratings of candidates, they git a ranked list for each user.

Their best performing run is " bupt_pris_2016_cs.2_.4_max", in which they put a higher weight on ratings from Foursquare (0.4), a lower weight on ratings from Yelp (0.2), and used a max function to calculate the users' rating for attractions.

5.3.5 UAmsterdam

UAmsterdam [13] studied contextual suggestion problem through neural user profiling and neural category preference modeling by the help of suggestions' endorsements being released by the TREC 2016 contextual suggestion track organizers. Their best performing run is "UAmsterdamDL", in which they studied how to predict relevant suggestions to the given user and context using category preference models.

In UAmsterdamDL, they cast the context-aware recommendation problem to a binary classification problem. In

order to learn a user preference model, they have used a deep neural network with 4 hidden layers having 478 units, in which 123 suggestion-category relevance features have been used as inputs of the network. In this model, for each user, preferences in the user's profile considered as a train set and suggestion candidates available in the phase 2 requests considered as the test set.

6. CONCLUSIONS

This section concludes our overview of the TREC 2016 contextual suggestion track. The track's main aim is the creation of a reusable test collections for the personalized POI recommendation task, which has proved a difficult task according to the previous studies [10, 12]. To this aim, we released the TREC CS web corpus, which is a crawl of the TREC contextual suggestion test collection. But fixing the test collection's content, we can overcome the dynamic nature of the contextual suggestion collection, and separate this effect from the personalization effects. We have also used a multi-depth pooling approach to improves reliability of the contextual suggestion systems scores based on measures at ranks deeper than the traditional pool cut-off. Moreover, we released attractions' endorsements being collected by NIST assessors, and participants showed considerable interest in using the endorsements to improve their contextual suggestion systems.

Acknowledgments

We thank all participants and the many volunteers who contributed to the test collections built over the last five years of the TREC Contextual Suggestion Track. This research is funded in part by the European Community's FP7 (project meSch, grant # 600851).

References

[1] M. Aliannejadi, I. Mele, and F. Crestani. Venue appropriateness prediction for contextual suggestion. In Voorhees and Ellis [17].

[2] M. Bayomi and S. Lawless. ADAPT_TCD: An ontology-based context aware approach for contextual suggestion. In Voorhees and Ellis [17].

[3] C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. Bias and the limits of pooling for large collections. *Information retrieval*, 10(6):491–508, 2007.

[4] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. M. Voorhees. Overview of the TREC 2012 contextual suggestion track. In E. M. Voorhees and L. P. Buckland, editors, *The Twenty-First Text REtrieval Conference Proceedings (TREC 2012)*. National Institute for Standards and Technology: NIST Special Publication 500-298, 2013.

[5] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, N. Simon, and E. M. Voorhees. Overview of the TREC 2013 contextual suggestion track. In E. M. Voorhees, editor, *The Twenty-Second Text REtrieval Conference Proceedings (TREC 2013)*. National Institute for Standards and Technology. NIST Special Publication 500-302, 2014.

[6] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. M. Voorhees. Overview of the TREC 2014 contextual suggestion track. In E. M. Voorhees and A. Ellis, editors, *Proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014)*. National Institute for Standards and Technology, NIST Special Publication 500-308, 2015.

[7] A. Dean-Hall, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. M. Voorhees. Overview of the TREC 2015 contextual suggestion track. In E. M. Voorhees and A. Ellis, editors, *Proceedings of the Twenty-Fourth Text REtrieval Conference (TREC 2015)*. National Institute for Standards and Technology, NIST Special Publication 500-319, 2016.

[8] M. Dehghani, J. Kamps, H. Azarbonyad, and M. Marx. Significant words language models for contextual suggestion. In Voorhees and Ellis [17].

[9] S. H. Hashemi and J. Kamps. Venue recommendation and web search based on anchor text. In *23rd Text REtrieval Conference (TREC)*, 2014.

[10] S. H. Hashemi, C. L. A. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. On the reusability of open test collections. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 827–830, 2015.

[11] S. H. Hashemi, C. L. A. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. Test collection building and maintenance in dynamic domains. In *15th Dutch-Belgian Information Retrieval Workshop (DIR)*, 2016.

[12] S. H. Hashemi, C. L. A. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. An easter egg hunting approach to test collection building in dynamic domains. In *Proceedings of NTCIR-EVIA 2016*, pages 1–8, 2016.

[13] S. H. Hashemi, N. O. Amer, and J. Kamps. Neural endorsement based contextual suggestion. In Voorhees and Ellis [17].

[14] G. Kalamatianos and A. Arampatzis. Recommending points-of-interest via weighted kNN, rated rocchio, and borda count fusion. In Voorhees and Ellis [17].

[15] M. Khorasani, H. Sadjadi, F. Ramazani, and F. Ensan. A context based recommender system through collaborative filtering and word embedding techniques. In Voorhees and Ellis [17].

[16] J. Mo, L. Lamontagne, and R. Khoury. Word embeddings and global preference for contextual suggestion. In Voorhees and Ellis [17].

[17] E. M. Voorhees and A. Ellis, editors. *Proceedings of the Twenty-Fifth Text REtrieval Conference (TREC 2016)*, 2017. National Institute for Standards and Technology, NIST Special Publication 500-321.

[18] D. Yin, S. Gao, Z. Peng, Y. Li, and R. Liu. Beijing university of posts and telecommunications (BUPT) at TREC 2016: A rating model based on tags for contextual suggestion. In Voorhees and Ellis [17].