

# Combining Opinion Profile Modeling with Complex Context Filtering for Contextual Suggestion

Peilin Yang and Hui Fang

Department of Electrical and Computer Engineering  
University of Delaware, USA  
{franklyn,hfang}@udel.edu

**Abstract.** In this paper we describe our effort on TREC 2015 Contextual Suggestion Track. Using opinions from online resources to model both user profile and candidate profile has been proven to be effective on previous TREC. This year we also leverage the power of building profile based on opinions. Opinions from well known commercial online resources are collected in order to build the profiles. Two kinds of opinion representations are used for the two submitted runs. Linear interpolation is leveraged to rank the candidate suggestions. Additionally, an advanced context filter which considers all possible factors such as trip type and trip duration is applied to the ranking results so that unwanted venues are removed from the final ranking list. Official results of our submitted runs show the effectiveness of the proposed method.

## 1 Introduction

TREC 2015 Contextual Suggestion Track provides researchers the chance to test their methods on providing better personalized suggestions. For this year’s track, the task changes in some way when comparing with previous tracks:

- There is a pre-task which aims to collect candidate suggestions from multiple sources. The candidate suggestions are served as the suggestion pool for the live experiment.
- There are separated tasks: live experiment and batch experiment. The live experiment is designed as a web service which responses to the suggestion requests. Top results from different participants in live experiment are collected to serve as the suggestion pool. Finally only 30 suggestions are sampled for the batch experiment. The task of batch experiment is to rerank the sampled suggestions. There are 211 requests in total for both live experiment and batch experiment.
- Several optional context requirements are added including the trip type, the trip duration, the type of group the person is travelling with and the season the trip will occur. Participants are expected to propose method which accommodates the context requirements.
- Description generation requirement is removed.

For this year’s TREC, we still rely on building user profiles to help ranking candidate suggestions. When building user profile, we utilize the opinions from online resource to enrich the profiles as what we did in the last year. For batch experiment, two review representations are examined for two submitted runs. Linear interpolation is used to compute the similarity between user profile and candidate profile. The ranking list is generated by sorting the similarity scores in descending order. In order to meet the complex contexts requirement proposed this year, an advanced context filter is applied to the ranking list so that the unwanted candidate suggestions are put to the end of the list.

## 2 Our Method

### 2.1 System Framework

Our method consists of the following parts:

- Useful information gathering,
- Profile modeling,
- Candidate suggestion ranking,
- Complex context filtering.

We will describe them in details in the following sub-sections.

## 2.2 Useful Information Gathering

We did not participate the pre-task which aims to collect potential candidate suggestions from participants. Instead, we gather the online opinions for the all the candidate suggestions which is similar with what we did previously [4, 2, 3]. This year we conduct a best-effort strategy to crawl online opinions in the following way: We first use the candidate suggestion name with its location (city + state) as the query to Google<sup>1</sup> it. We extract the search result pages belong to Yelp<sup>2</sup>, TripAdvisor<sup>3</sup> and OpenTable<sup>4</sup> from the first 50 results. We then collect the information from the above mentioned web sites in order to get as many as possible opinions.

We call the above mentioned strategy as the best-effort strategy because sometimes the names of the candidates are not well organized as it should be. For example, the name “Mobile Home Parks near Watertown , NY : 153 Listed” in the official collection file could be better changed to ”Watertown”. Moreover, there are some unrelated venues accidentally appear in the first 50 search results returned by Google either because of their similar name or other factors which are also included as the information sources of the candidate.

After crawling, information including the category of the suggestion, the number of reviews, average rating, business hour, price range, all ratings and the associated text reviews of the candidate suggestion are extracted and kept in JSON format for further use. Finally, approximately 161,907 candidate suggestions are crawled out of 1,235,844 candidate suggestions in total.

## 2.3 Profile Modeling

We use opinion to model user profile as well as candidate suggestion profile due to its richness. Specifically, we use positive opinions of the suggestions that the user likes (positive suggestions in preferences) to build her positive user profile, and use negative opinions of suggestions that the user dislikes (negative suggestions in preferences) to build negative user profile. The intuition is that users with similar ratings of a suggestions share something in common of why they like or dislike the suggestion.

Formally, the user profiles are estimated as [2]:

$$\mathcal{U}_{pos} = \bigcup_{reqs_i \in ES(U) \cap R_U(reqs_i)=POS} REP(O_{pos}(reqs_i)) \quad (1)$$

$$\mathcal{U}_{neg} = \bigcup_{reqs_i \in ES(U) \cap R_U(reqs_i)=NEG} REP(O_{neg}(reqs_i)) \quad (2)$$

where  $reqs_i$  is the  $i$ th suggestion in preferences  $i$ .  $O_{pos}(reqs_i)$  represents all positive text reviews about  $reqs_i$ ,  $O_{neg}(reqs_i)$  represents all negative text reviews, and  $REP(O(reqs_i))$  denotes how to represent text reviews  $O(reqs_i)$  in the profile.  $R_U(reqs_i)$  is the rating of example suggestion  $reqs_i$  given by user  $U$ . The original value of  $R_U(reqs_i)$  could be numerical, and we map these values into either POS or NEG. We build the positive and negative profile for a candidate suggestion similar to what we did in building user profile as follows:

$$\begin{aligned} \mathcal{CS}_{pos} &= REP(O_{pos}(CS)) \\ \mathcal{CS}_{neg} &= REP(O_{neg}(CS)). \end{aligned}$$

For user ratings in Contextual Suggestion Track collection, we map the example suggestions with rating  $\{3, 4\}$  from the user as positive and opinions with rating  $\{0, 1\}$  from the user as negative. Example suggestions with rating  $\{2\}$  are simply ignored since they are hard to categorize into either positive or negative. For the opinions crawled from Yelp, TripAdvisor and OpenTable (either for example or candidate), opinions with rating  $\{4, 5\}$  are viewed as positive and opinions with rating  $\{1, 2\}$  are treated as negative. Opinions with rating  $\{3\}$  are simply ignored and are not used for building the profiles since they often contain mixed polarities and thus are hard to categorize.

We tried two strategies of how to represent the profiles.

- *Use full reviews (FR)*: use full text in the review.
- *Use nouns (NR)*: nouns in the review.

Simple pre-processing are performed on the original text reviews: 1) Terms are lower cased; 2) Stop words are removed.

<sup>1</sup> <https://www.google.com>

<sup>2</sup> <http://www.yelp.com>

<sup>3</sup> <http://www.tripadvisor.com>

<sup>4</sup> <http://www.opentable.com>

## 2.4 Candidate Suggestion Ranking

Given a user  $U$  and a candidate suggestion  $CS$ , the similarity score can be computed as follows:

1. Build a positive and negative user profile, i.e.,  $\mathcal{U}_{pos}$  and  $\mathcal{U}_{neg}$ , based on the information about example suggestions that the user has rated, i.e.,  $ES(U)$ ;
2. Build the positive and negative profile for the candidate suggestion  $CS$ , i.e.,  $\mathcal{CS}_{pos}$  and  $\mathcal{CS}_{neg}$ ;
3. Predict the score of  $CS$ , i.e.,  $S(U, CS)$ , based on the similarities between  $\mathcal{U}_{pos}$ ,  $\mathcal{U}_{neg}$ ,  $\mathcal{CS}_{pos}$  and  $\mathcal{CS}_{neg}$ .

We use linear interpolation to compute  $S(U, CS)$ . In previous TREC, we have also tried learning to rank (LTR) as the ranking method. However, linear interpolation is still favorable for following reasons:

1. It fits the requirement of live experiment. LTR often needs time to train the model which basically not applicable for real time service. Linear interpolation can compute the score really quick.
2. We did not get very promising performance using LTR in previous TREC.

The equation used to estimate the similarity between a user and a candidate suggestion is as follows:

$$S(U, CS) = \alpha \times SIM(\mathcal{U}_{pos}, \mathcal{CS}_{pos}) - \beta \times SIM(\mathcal{U}_{pos}, \mathcal{CS}_{neg}) - \gamma \times SIM(\mathcal{U}_{neg}, \mathcal{CS}_{pos}) + \eta \times SIM(\mathcal{U}_{neg}, \mathcal{CS}_{neg}) \quad (3)$$

where  $SIM(a, b)$  is text similarity measurement between any two profiles.  $\alpha, \beta, \gamma, \eta \in [0, 1]$  are parameters that balance the impact of the different similarities to the final similarity score. We use an axiomatic similarity function F2EXP [1] when computing  $SIM(a, b)$ . A preliminary training process was done on our self-crawled Yelp data collection<sup>5</sup>. The trained parameters set is finally as  $\alpha = 1.0, \beta = 0.0, \gamma = 0.9, \eta = 0.1$ .

## 2.5 Context Filter

This year’s track has a noticeable change comparing with previous tracks: The context is much more complex. Specifically, trip type, trip duration, type of group the person is travelling with, season of the trip are added to the context requirement. To better solve the context constraints of the problem, we propose an advanced context filter which can remove (actually it just removes the unwanted suggestions at the end of the ranking list) the unwanted suggestions if they do not meet with the context requirements.

Our context filter mainly has three kinds of operations: *Boost*, which boost the score of the suggestion; *Avoid*, which remove the suggestion; *Mix*, which basically pick suggestions from different categories in a round-robin fashion so that the diversity of the ranking list increases. Other information that is also used in the context filter includes the price, the category and business hours etc.

For trip type, there are three possible values: *Business*, *Holiday*, *Other*. For business, we simply boost the pricey hotels and restaurants. For holiday and other trip type the context filter just does nothing.

The possible trip durations are: *Night Out*, *Day Trip*, *Weekend Trip*, *Longer*. For night out, we boost bar, pub, theaters, music venues and avoid venues that are closed at night, e.g. brunch restaurants. For day trip, in contrast we avoid hotel, bar, pub, theaters and also avoid closed venues. For weekend trip, we mix hotels, restaurants and landmarks for every 5 suggestions in the ranking list. For longer trip, we mix hotels with other types of venues for every 5 suggestions in the ranking list.

The trip group includes: *Travelling Alone*, *Travelling with a group of friends*, *Travelling with family*, *Travelling with an other group*. For family trip we boost the amusement park. For group trips we boost the venues which bear “good for groups” property (Yelp and OpenTable have such information).

For season of the trip, we just avoid park, amusement park and zoo for winter.

## 3 Submitted Runs and Experiment Results

In this section, we report the results for batch experiment. We submitted two runs: UDInfoCS2015\_fr and UDInfoCS2015\_nr. UDInfoCS2015\_fr uses FR as the review representation and UDInfoCS2015\_nr uses NR as the review representation. For both runs, we apply the advanced context filter to the ranking list.

<sup>5</sup> Available at [https://s3.amazonaws.com/irj2014\\_yelp\\_data/irj2014\\_yelp.tar.gz](https://s3.amazonaws.com/irj2014_yelp_data/irj2014_yelp.tar.gz).

**Table 1.** Overall Mean Performances

Runs	fr	nr	fr-no-cf	nr-no-cf
P@5	0.5583	0.5507	0.5972	0.6038

**Table 2.** Statistics of reviews

	reviews cnt. (mean)	reviews cnt. (std)	pos. terms cnt. (mean)
All Candidates	709	1650	46876
Relevant Candidates	938	1943	62288
Non-Relevant Candidates	491	1275	31977
fr	1217	1999	78586
fr-no-cf	1279	2080	81181
nr	1176	2017	76234
nr-no-cf	1183	2019	75390

Table 1 shows the overall mean performances of our runs in terms of P@5. We also include the results of not applying the context filter as “-no-cf”. We can see from the table that both of our runs have very promising performances. This confirms the effectiveness of using opinions to rank candidate suggestions. However, using context filter can hurt the performance a little bit as the performances with context filter are better. We may need to design more effective context filter in the future work. Since we have participated three years TRECs in a row, it is worthy to analyze more about our method since the main idea is highly consistent.

One important statistic we would like to evaluate is the number of reviews in both candidates and in our results. Our method in natural favors the suggestions with at least some amount of reviews (terms), especially the positive reviews. Table 2 shows the detailed statistics about the number of the reviews for both candidates and our results. From the table we can see that for our results the average number of reviews in the top 5 results is around 1,200 which is much larger than the average number (709) of reviews in the candidates. Even the number of reviews for relevant candidates is larger (average 938) the gap between it and our results is relatively huge. Moreover, we also have the statistic for number of terms in the reviews. This is a more reflective factor to look as the terms are directly used by our method. From the table we see that the average number of terms in the review for our results is much larger than that of for all the candidate suggestions. The gap between our results and the relevant candidate suggestions is also huge (approximately 15,000 terms difference). The quantities listed may unveil the shortage of our method and we would like to do more investigations about how the statistics of the reviews correlates with the performance in the future work.

## 4 Conclusion

In Contextual Suggestion Track 2015, we leverage opinions as the source to model user and candidate profile. We use linear interpolation method as the ranking method which essentially computes the similarity between user profile and candidate profile. This year we also propose an advanced context filter which boost or remove the suggestions in the ranking list so that we can meet the context requirement. We submit two runs to Contextual Suggestion Track 2015. In general, both of them have promising performances. We further analyze the advantage and disadvantage of our method in order to provide more investigation results to the research community.

## References

1. H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 480–487, New York, NY, USA, 2005. ACM.
2. P. Yang and H. Fang. An opinion-aware approach to contextual suggestion. In *Proceedings of TREC'13*, 2013.
3. P. Yang and H. Fang. Exploration of opinion-aware approach to contextual suggestion. In *Proceedings of TREC'14*, 2014.
4. P. Yang, H. Wang, H. Fang, and D. Cai. Opinions matter: A general approach to user profile modeling for contextual suggestion. *Inf. Retr.*, 18(6):586–610, Dec. 2015.