

# Opinions matter: a general approach to user profile modeling for contextual suggestion

Peilin Yang<sup>1</sup> · Hongning Wang<sup>2</sup> · Hui Fang<sup>1</sup> · Deng Cai<sup>3</sup>

Received: 7 March 2014 / Accepted: 28 October 2015 / Published online: 19 November 2015  
© Springer Science+Business Media New York 2015

**Abstract** The increasing use of mobile devices enables an information retrieval (IR) system to capitalize on various types of contexts (e.g., temporal and geographical information) about its users. Combined with the user preference history recorded in the system, a better understanding of users' information need can be achieved and it thus leads to improved user satisfaction. More importantly, such a system could *proactively* recommend suggestions based on the contexts. User profiling is essential in contextual suggestion. However, given most users' observed behaviors are sparse and their preferences are latent in an IR system, constructing accurate user profiles is generally difficult. In this paper, we focus on location-based contextual suggestion and propose to leverage users' opinions to construct the profiles. Instead of simply recording "what places a user likes or dislikes" in the past (i.e., description-based profile), we want to construct a profile to identify "why a user likes or dislikes a place" so as to better predict whether the user would like a new candidate suggestion of place. By assuming users would like or dislike a place with similar reasons, we construct the opinion-based user profile in a collaborative way: opinions from the other users are leveraged to estimate a profile for the target user. Candidate suggestions are represented in the same fashion and ranked based on their similarities with respect to the user profiles. Moreover, we also develop a novel summary generation method that utilizes the opinion-based user profiles to generate personalized and high-quality

---

✉ Peilin Yang  
franklyn@udel.edu

Hongning Wang  
hw5x@virginia.edu

Hui Fang  
hfang@udel.edu

Deng Cai  
dengcai@cad.zju.edu.cn

<sup>1</sup> Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, USA

<sup>2</sup> Department of Computer Science, University of Virginia, Charlottesville, VA, USA

<sup>3</sup> The State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China

summaries for the suggestions. Experiments are conducted over three standard TREC contextual suggestion collections and a Yelp data set. Extensive experiment comparisons confirm that the proposed opinion-based user modeling outperforms the existing description-based methods. In particular, the systems developed based on the proposed methods have been ranked as top 1 in both TREC 2013 and 2014 contextual suggestion tracks.

**Keywords** Contextual suggestions · Opinions · User modeling · Recommendation

## 1 Introduction

The increasing availability of internet access on mobile devices, such as smart phones and tablets, has made mobile search a new focus of information retrieval (IR) research community. The contextual information such as geographical and temporal information that is available in mobile search environment provides unique opportunities for IR systems to better understand its users. Moreover, a user's preference history collected in a mobile search system can be incorporated with such contextual information to better understand the user's informational need. Ideally, a mobile search system should thus *proactively* generate suggestions for various user information needs. For example, it would be useful to automatically send recommendations about the Beatles museum to a music fan who travels to Liverpool. In addition to returning a list of suggestions to the user, it would also be useful to provide a short yet *informative summary* for each suggestion so that the user can easily decide whether the recommended suggestion is interesting before accepting it. This problem is referred to as *contextual suggestion*, and has been identified as one of the IR challenges (i.e., “finding what you need with zero query terms”) in the SWIRL 2012 workshop (Allan et al. 2012).

In this paper, we focus on the problem of location-based contextual suggestion. There are two necessary steps to perform the location-based contextual suggestion: (1) identify a set of candidates that satisfy the contextual requirements, e.g., places of interest that are close to a target place; (2) rank the candidates with respect to the user interest. Here we assume that the candidates from the first step are given, and we focus on developing effective solution for the second step.

User profiling is essential to effectively rank candidate places with respect to a user's information need. Previous studies explored the category and description information about places to construct user profiles (Yang and Fang 2012). We argue that only using category or description to build a user profile is not sufficient: category of places is too general to capture a user's underlying needs; while the text description of a place is too specific to be generalized to other places.

To address this difficulty, in this paper, we propose to leverage opinions, i.e. opinion ratings and the associated text reviews, to construct an opinionated user profile, which aims to explain “why the user likes or dislikes the suggestions” instead of simply recording “what places the user liked or disliked” in the search history. However, users seldom share their opinions for every place they visited before. The lack of user-specific opinion data is the main obstacle in building an opinion-based user profile. To address the data sparsity challenge, we propose to build the user profile by leveraging opinions from the other users. Our basic assumption is that users with similar preferences of place suggestions would have similar reasons to like or dislike the place. A user profile is then

divided into a positive part and a negative part. When a user likes a suggestion, the positive profile of this user is enhanced by the opinions about this place from similar users, i.e., those who also liked the place. Accordingly, the negative profile is enhanced in a similar way, i.e., leveraging opinions from other users who did not like the place. The candidate place suggestions are modeled in the same fashion: positive and negative opinions of users towards the candidate suggestion are aggregated accordingly as the positive and negative profile of the candidate suggestion. The ranking of candidate suggestions is decided by the similarities between such opinion-based candidate profile and user profile. More specifically, we explore four ways of representing profiles based on the suggestions and different strategies to combine profile similarities for candidate ranking.

In order to provide users with more informative summary for each candidate suggestion, we also propose a structured summary generation method that incorporates information from multiple resources, such as the candidate suggestion's website and its online reviews. Each summary includes different aspects about a suggestion, e.g., type of the place, a short description about the place, how other users commented about this place, and why this place is recommended to given user. These aspects are expected to highlight important information about the candidate suggestion that is most concerned by users.

We conduct experiments to evaluate the proposed candidate ranking methods based on standard TREC contextual suggestion collections as well as a collection crawled from Yelp. Results show that the proposed opinion-based user profiling is more effective than the category or description-based baseline methods. The proposed opinion-based method is more robust when fewer data are used to construct user profiles. Finally, two sets of experiments based on user studies show that the proposed method can generate more useful summaries for candidate suggestions.

The rest of the paper is organized as follows. Section 2 formalizes the problem of contextual suggestion. Section 3 describes the proposed opinion-based user profile modeling. Based on the opinion-based profile modeling, we discuss how to rank candidate suggestions in Sect. 4, and how to generate informative summaries for each candidate in Sect. 5. Finally, we present experiment results in Sect. 6 and conclude in Sect. 8.

## 2 Problem formulation

The problem of contextual suggestion can be formalized as follows. Given a user's contexts (e.g., location and time) and the her/his preferences on a few example suggestions, the goal is to retrieve candidate suggestions that can satisfy the user's information need based on both the context and preferences. For each returned candidate suggestion, a short description may also be returned so that the user could decide whether the suggestion is interesting without going to its website. For example, assume that a user liked "Magic Kingdom Park" and "Animal Kingdom", but disliked "Kennedy Space Center". If the user is visiting Philadelphia on a Saturday, the system is expected to return a list of suggestions such as "Sesame Palace" together with a short summary of each suggestion, e.g., "Sesame Place is a theme park in Langhorne, Pennsylvania based on the Sesame Street television program. It includes a variety of rides, shows, and water attractions suited to very young children."

Since our paper focuses on user modeling, we assume that we have filtered out the suggestions that do not meet the context requirement and the remaining suggestions only

need to be ranked based on the relevance to user preferences. Note that the filtering process based on contexts can be achieved by simply removing the suggestions that do not satisfy the contextual requirements, such as the ones that are either too far away from the current location or those that are currently closed.

The remaining problem is essentially a ranking problem, where candidate suggestions need to be ranked based on how relevant the suggestions are with respect to a user's interest. Formally, let  $U$  denote a user and  $CS$  denote a candidate suggestion, we need to estimate  $S(U, CS)$ , i.e., the relevance score between the user and the suggestion.

It is clear that the estimation of the relevance score is related to how to represent  $U$  and  $CS$  based on the available information. Let us first look at what kind of information we can gather for  $U$  and  $CS$ . For each user  $U$ , we know the user's preferences (i.e., ratings) for a list of example suggestions. We denote an example suggestion  $ES$  and its rating given by user  $U$  as  $R(U, ES)$ . For a suggestion (either  $CS$  or  $ES$ ), we assume that the following information about the suggestion is available: the text description such as title and category and online opinions about this suggestion. Note all the information can be collected from online location services such as Yelp and Tripadvisor.


### 3 Opinion-based user profile modeling

#### 3.1 Basic idea

The problem of user profile modeling is to study how to represent  $U$  with all the available information. In our problem setup, the available information for a user  $U$  includes the user's preferences for a set of example suggestions. Existing studies often estimated user profiles based on the descriptive information of the example suggestions such as their names, descriptions and web sites (Bellogín et al. 2013; Hubert and Cabanac 2012; Koolen et al. 2013; Rao and Carterette 2012; Yang and Fang 2012, 2013a, b). However, one limitation of this approach is that such descriptive information could be very specific for one suggestion and might not be useful at all to infer the user's preferences on other suggestions. Categories of the suggestions were then used by some methods to overcome the limitation (Roy et al. 2013; Yang and Fang 2012; Yates et al. 2012). Although this method improves the performance, the improvement is often limited since category information might be too general to capture the reasons behind the user preferences.

Instead of simply capturing what a user likes or dislikes, i.e. the descriptive information of example suggestions, we propose to model the user profile based on the user's opinions about the example suggestions. The opinions about a suggestion is defined as the (rating, review text) pairs in our paper. When determining whether an opinion is positive or negative, we rely on the numeric rating rather than the review text. More details about this are described in Sect. 6.2.1.

We now motivate the opinion-based user modeling through an example as shown in Fig. 1. Assume that we know a user's preferences for the first four suggestions and want to infer the user preference for the last one. Neither description-based nor category-based methods are effective here. For example, the category of the candidate suggestion is "hotel", which does not match with the categories of all the example suggestions. Moreover, the descriptions of these example suggestions are very specific, making it

	Category	Description (web site)	Review	Preference
	Museum	The A Museum is the oldest Holocaust museum in the United States...	A small and <b>clean</b> museum that will take you less than an hour to see everything...	✓
	Hotel	The B Hotel is just moments from all tourists attractions and exciting things to do in Los Angeles both for business and pleasure..	<b>Dirty</b> hotel, the room itself was filthy...	✗
	Restaurant	The ambiance at C is palpable. Inside our old roadhouse, you feel like you are back in the old west with our long, long “did I say” long bar....rustic décor and welcoming taff. Makes you feel right at home the minute you walk in the door... warm and friendly like!	“Good food, <b>clean</b> restaurant” - My daughter and I enjoyed the corn dog... Women's bathroom was very <b>clean</b> , much appreciated.	✓
	Food	Country-style comfort food including all-day breakfasts & hearty lunches served in a homey space.	Awful in every conceivable way. Bad service, <b>dirty</b> environment, and tasteless slop. 2 stars for a sort of decent beer selection.	✗
	Hotel	Hotel Z features an outdoor pool for hotel guests only and indoor/outdoor private event space...	Great hotel! <b>clean</b> and modern...	?

**Fig. 1** An example scenario when we know the user’s preferences for some suggestions and want to predict the preference for the unknown one

difficult to find their commonalities. However, if we are able to know the user’s preference and review for each example suggestion, it would be possible for us to more accurately infer why the user liked or disliked these places. For example, it seems that the two suggestions that the user liked (i.e., example suggestions 1 and 3) are “clean” while the places that the user disliked (i.e., example suggestions 2 and 4) are both “dirty”. Thus, we may infer that the user prefers places that are “clean”. Now if we know that a candidate suggestion is well known for its “cleanness” based on online reviews, we could safely infer that the user would like this candidate suggestion. Clearly, opinion- based user profile modeling should be more effective than the category- based and description-based methods since it can capture user preferences more accurately.

One challenge of using opinions to model user profile is that users may not share their opinions explicitly by writing the reviews for each example suggestion. To address the challenge, we propose to leverage opinions from similar users. More specifically, we

assume that users who rate a suggestion similarly would share the similar opinions about the suggestion. If a user likes a suggestion, we could identify all other users who also like this suggestion and leverage their reviews about the suggestion as part of the user’s positive profile, i.e., the profile about what the user likes. We can build the negative profile in a similar way.

Specifically, we use positive reviews of the example suggestions that the user likes to build his or her positive user profile, and use negative reviews of example suggestions that the user dislikes to build negative user profile. The basic assumption is that the opinion of a user about a place can be inferred by the opinions of the users who share the same preference as the target user to the same place.

Formally, a user  $U$ ’s positive profile  $\mathcal{U}_+(U)$  can be estimated as follows:

$$\mathcal{U}_+(U) = \bigcup_{\forall i, R(U, ES_i)=POS} REP_+(ES_i), \tag{1}$$

where  $ES_i$  is an example suggestion and  $R(U, ES_i)$  is the rating of  $ES_i$  given by user  $U$ . The ratings could be binary or within a specified range, but they can be mapped to either positive (i.e., *POS*) or negative (i.e., *NEG*). We will provide more details on these mappings in our experiment setup.  $REP_+(ES_i)$  is the positive opinion based representation for  $ES_i$  and we will provide more details about the representation in the following subsection (i.e., Sect. 3.2).

Similarly, a user  $U$ ’s negative profile  $\mathcal{U}_-(U)$  can be estimated as:

$$\mathcal{U}_-(U) = \bigcup_{\forall i, R(U, ES_i)=NEG} REP_-(ES_i), \tag{2}$$

where  $REP_-(ES_i)$  is the negative opinion based representation for  $ES_i$ .

### 3.2 Opinion-based representation for suggestions

We now discuss how to generate opinion-based representations for the suggestions (*CS* or *ES*). Given an *ES*, we need to construct two profiles: (1) positive profile, i.e.,  $REP_+(ES)$ , based on all the positive reviews of *ES*; and (2) negative profile, i.e.,  $REP_-(ES)$  based on all negative reviews of *ES*.

Now the remaining challenge is how to construct these two profiles based on the reviews. For example, do we include every term from the reviews? Or shall we only include important terms from the reviews? If so, how to select the important terms and what are the impact of the selected terms? In order to answer all these questions, we explore the following four strategies to construct  $REP_+(ES)$  and  $REP_-(ES)$  based on the reviews. All of these strategies are based on “bag-of-terms” representations but they are different in which terms from the reviews are used in the representations.

- *Full reviews (FR)* The simplest approach is to take all terms occurring in the review text to build the profile. For example, when estimating  $REP_+(ES)$ , we take all the positive reviews about *ES* and use bag of terms representations for these reviews. We can estimate  $REP_-(ES)$  in a similar way using negative reviews. Despite its simplicity, this representation may cause the efficiency concern because when more reviews are available, the size of the profiles could be fairly large.
- *Selective term based reviews (SR)* To reduce the computational cost, one possibility would be to construct the profile based on a set of selected terms. Terms could be selected using different criteria, and we include the most frequent terms in the profiles.

Specifically, top 100 most frequent terms in the review text are selected and their frequencies are set to 1 after being selected. This strategy would be less computational expensive than the FR method, but it may not perform as well since using only frequent terms might not be the best way of representing opinions.

- *Noun based reviews (NR)* Another strategy that we have explored to generate concise profiles based on reviews is to only use the nouns from the review text. The rationale is that nouns often correspond to important aspects of a suggestion, and nouns are less noisy than the frequent terms. Thus, we expect better performance of this method compared with *SR*.
- *Review summaries (RS)* Finally, we leverage the Opinosis algorithm (Ganesan et al. 2010), an unsupervised method that generates concise summaries of reviews, to construct the profiles. The algorithm first generates a textual word graph (called the Opinosis-Graph) of the input data, where each node represents a word, and an edge represents the link between two words. Using three unique properties of the graph data structure (redundancy capture, collapsible structures, gapped sub-sequences), various promising sub-paths in the graph that act as candidate summaries are scored and ranked. The top candidate summaries are then used to generate the final Opinosis summaries. In this work, we first concatenate all the reviews and then generate the review summary using the Opinosis algorithm.

Figure 2 shows an example of the original review and the results of different opinion-based representations. When building user profile models, we perform the following simple pre-processing on the original reviews: (1) converting terms into lower cases; and (2) removing punctuations and stop words.

**Original Review:**

Funky little spot with a *laid-back vibe* and good chow. The chile sauce had plenty of flavor and kick, and everything seemed fresh. Service was friendly and reasonably quick, and the prices were reasonable. A bit expensive but *great food* and a great ambiance. I had the club sandwich with green chile and it was delicious. Very, very good. Party of 6 - huevos rancheros, veggie burrito, steak tacos, Mac and cheese with *green chile*. Topped off by *key lime pie*. All servings enjoyed by all. If you want ambience skip. If you want a quick, good, no frills meal, this place is for you. The *best Mexican food* I've had in a long time. The Blue Corn Enchiladas with Green Chilis were fantastic.

**FR:**

*The same as the the raw opinion sentences above except with removal of stop words.*

**SR:**

chile want vibe veggie time tacos steak spot skip servings seemed sauce sandwich reasonably reasonable rancheros prices plenty place pie off no meal long huevos...

**NR:**

chow sauce plenty flavor kick everything Service prices bit food ambiance club sandwich chile Party burrito steak tacos Mac cheese chile lime pie servings...

**RS:**

best santa fe; green chile; key lime pie; great food; back vibe.

**Fig. 2** An example results of different opinion-based representations

### 4 Candidate suggestions ranking

We now describe how to rank candidate suggestions based on the user profiles. As described in the previous section, we can estimate a user’s profile based on the user’s preferences on the example suggestions as well as the reviews of the example suggestions. In particular, the profile of user  $U$  can be represented with  $\mathcal{U}_+(U)$  and  $\mathcal{U}_-(U)$ . Similarly, a candidate suggestion  $CS$  can be represented based on its positive and negative reviews, i.e.,  $REP_+(CS)$  and  $REP_-(CS)$ . Thus, the relevance score  $S(U, CS)$  should be related to the similarities between the positive/negative user profiles and the positive/negative representations of candidate suggestions.

In order to compute  $S(U, CS)$ , we investigate two possible ways of combining these similarities: linear interpolation and learning-to-rank.

#### 4.1 Linear interpolation

Linear interpolation is a simple yet effective method to combine multiple scores into one. The main idea here is to linearly combine the similarity scores between user profiles (i.e.,  $\mathcal{U}_+(U)$ ,  $\mathcal{U}_-(U)$ ) and the candidate profiles (i.e.,  $REP_+(CS)$  and  $REP_-(CS)$ ).

In the previous section, we have discussed how to construct these profiles, now we discuss how to compute their similarities. Our basic idea is illustrated in Fig. 3. Intuitively, a user would prefer suggestions with the properties that the user likes or those without the properties that the user dislikes. This means that the relevance score  $S(U, CS)$  should be positively correlated with the similarity between two positive profiles and two negative profiles, i.e.,  $SIM(\mathcal{U}_+(U), REP_+(CS))$  and  $SIM(\mathcal{U}_-(U), REP_-(CS))$ . Similarly, a user would not like suggestions with the properties that the user dislikes or suggestions without the properties that the user likes, which means  $S(U, CS)$  should be negatively correlated with the similarity between positive and negative profiles, i.e.,  $SIM(\mathcal{U}_+(U), REP_-(CS))$  and  $SIM(\mathcal{U}_-(U), REP_+(CS))$ .

Following the above intuitions, we can estimate the similarity between a user and a candidate suggestion as follows:

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

where  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\eta$  are parameters that balance the impact of the four components to the final similarity score. All of their values are between 0 and 1.  $SIM(a, b)$  could be any text

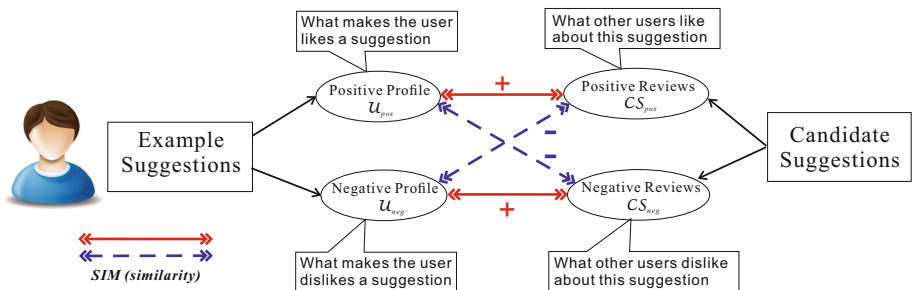


Fig. 3 The linear interpolation method



similarity measure. In this paper, we used an axiomatic retrieval function F2EXP (Fang and Zhai 2005) since it has been shown to be effective for long queries (Fang and Zhai 2005). So, we have

$$SIM(a, b) = \sum_{t \in a \cap b} \frac{c(t, b)}{c(t, b) + 0.5 + \frac{0.5 \cdot |b|}{avdl} \cdot \left(\frac{N+1}{df(t)}\right)^{0.35}} \quad (4)$$

where  $c(t, b)$  is the occurrences of term  $t$  in  $b$  and  $|b|$  is the number of terms in  $b$ .  $avdl$  is the average length of all the candidate suggestion representations,  $N$  is the number of candidate suggestions in the collection, and  $df(t)$  is the number of candidate suggestion representations that contain term  $t$ . Note that there are two collections for the candidate suggestion representations, i.e., positive one versus negative one. Depending on whether  $b$  is a positive or negative representation, the last three statistics are computed based on the corresponding collection.

## 4.2 Learning to rank

Machine learning is another way of combining multiple features. And learning to rank has been proven to be effective in information retrieval area (Liu 2009; Macdonald et al. 2013).

For our task, we can first compute the similarity scores  $SIM(\mathcal{U}_+(U), REP_+(CS))$ ,  $SIM(\mathcal{U}_-(U), REP_-(CS))$ ,  $SIM(\mathcal{U}_+(U), REP_-(CS))$  and  $SIM(\mathcal{U}_-(U), REP_+(CS))$  which is exactly the same as what we do in linear interpolation method (Sect. 4.1). After having these similarities at hand, we can use the similarities as features and use learning-to-rank methods to compute the ranking score for each candidate suggestion. The following learning-to-rank methods are considered:

- *MART*, which is also known as Gradient Boosted Regression Trees. It generates a set of weighted regression trees that aim to predict the scores of training data (Friedman 2000). The regression tree learned at each iteration only needs to focus on the difference between the target label and the prediction of previous trees. The number of trees can be tuned via the validation data.
- *LambdaMART*, which also applies boosted regression trees, but the training of the trees consider numeric measurements (such as NDCG and ERR) to obtain the gradient of the surrogate loss function between pairs of documents (Burges 2010). Like MART, the number of iterations can also be tuned via the validation data. It is denoted as LMART in the paper.
- *LinearRegression*, which views the target label as a linear combination of the attributes. The goal is to search for parameters so that the sum of the squares of differences between target label and the predicted label is minimized. It is denoted as LR in the paper.

## 5 Structured summary generation

Here we discuss how to generate a *personalized* and *structured* summary for a candidate suggestion. A straightforward solution is to apply existing text summarization techniques and extract important information from the website of a suggestion (Dean-Hall et al.

2013). The result would be similar to the search snippets generated by Web search engines for the suggestion's website. For example, the snippet of *Olive Room*<sup>1</sup> is “The Olive Room, French Restaurant in Montgomery. See the menu, 49 photos, 1 blog post and 34 user reviews. Reviews from critics, food blogs and fellow...”

Although this strategy would work, it might not be optimal for the following reasons. First, the summary comes from only a single information source, i.e., the website of the suggestion, which may lead to incomplete or even biased information about the suggestion. Second, the summary is not personalized. The lack of personalization might not effectively convince every user.

To overcome these limitations, we propose a novel summarization method for contextual suggestions that leverages the user profile as well as the information from multiple sources about the suggestions to produce *personalized* and *structured* summaries.

Given a suggestion, we could collect a wide variety of information about the suggestion, which includes the category of the suggestion, website of the suggestion as well as the reviews of the suggestion. Note that the category and reviews of a suggestion can be downloaded from the third party websites such as Yelp and Tripadvisor. Recall that the user profiles we have estimated can tell us what makes a user like or dislike a suggestion. Thus, it would be interesting to study how to leverage user profiles to generate summaries that are more convincing. Now, the key challenge is how to synthesize the information from various sources and generate a coherent personalized summary.

To tackle this problem, we propose to generate a structured summary. In particular, the summary consists of multiple fields, and each field aims to provide information about a unique aspect of the suggestion. All the fields together would offer a more complete information about the suggestion as well as arguments on why the suggestion would be appealing to a particular user.

The structured summary consists of the following four components:

- *An Opening Sentence* It provides a high-level introduction in one sentence.
- *An “official” introduction* It provides more detailed information about the suggestion by extracting information from the website of the suggestion.
- *Highlighted reviews* This component explains why other people like this suggestion based on the information extracted from the reviews.
- *A concluding sentence* This component explains why this suggestion is recommended to the user.

We now provide more detailed information on how to generate the above structured summary.

*An opening sentence* The opening sentence serves as a high-level introduction sentence. Sometimes people can even hardly know what kind of the suggestion it is by looking at its name. For instance, we might guess that “Zahav” is related to food, but what kind of food? Intuitively, the opening sentence should clearly explain what this suggestion is. And the category information of this suggestion could be a good choice. Our opening sentence then is of the form: suggestion's name followed by the fine category of that suggestion. For example, “The big fish grocery is a *shopping store especially for seafood*.” If the fine category of candidate suggestion is not available, we show its coarse category like “The DCM is a museum.” The fine and coarse category can be obtained from the data sources such as Yelp and Google Place.

---

<sup>1</sup> <http://www.theoliveroom.com>.

*The “official” introduction* The “official” introduction consists of useful sentences extracted from the web site of the suggestion. Generally speaking, we cannot rely on the HTML DOM structure to extract the well crafted description for two reasons: (1) there might not be dedicated field to store such information, even in the meta data; (2) even if we can find a short summary in the meta data, the information might be too general and does not match user interests well. To address this challenge, we propose to leverage reviews to identify important information from the websites. Specifically, we first extract nouns with high frequency from the suggestion opinions. After that, we use these nouns to identify the sentences from the web site of the candidate suggestion. All the identified sentences are ranked based on the number of distinctive/total positive adjectives. Only top 5 ranked sentences are used due to the length of the summary.

*The highlighted reviews* The highlighted reviews are the sentences extracted from the positive reviews of the suggestion. The process is very similar with the extraction of “official” introduction. We use the most frequent nouns as a guide to extract sentences from positive reviews. Sentences with more distinct positive adjectives are chosen.

*The concluding sentence* The concluding sentence is the last sentence in the structured description. Here we customize it to specific user. The concluding sentence is of the form: “We recommend this suggestion to you because you liked *abc* and *xyz* in example suggestions.” *abc* and *xyz* are example suggestions that have the same fine category as the candidate suggestion.

As an example, here is the generated summary for a candidate suggestion, i.e., *Olive Room*. “*The Olive Room is a bar. HERE ARE THE DESCRIPTIONS FROM ITS WEBSITE: Here at the olive room, you will receive the finest cuisine montgomery has to offer, hands down. HERE ARE REVIEWS FROM OTHER PEOPLE: If you are looking for a unique dining experience, with excellent food, service, location, and outstanding ambiance, look no further! THIS PLACE IS SIMILAR TO OTHER PLACE(S) YOU LIKED, i.e. Tria Wine Room.*”

## 6 Experiments

We conduct experiments to evaluate the proposed opinion-based candidate ranking methods as well as the summarization method.

### 6.1 Data sets

To evaluate the effectiveness of the proposed methods, we conduct experiments over two types of data sets: (1) the data set used in the TREC Contextual Suggestion track Dean-Hall et al. (2012); and (2) a data set crawled from Yelp.<sup>2</sup>

- *TREC data set* The TREC Contextual Suggestion Track Dean-Hall et al. (2012) provides an evaluation platform for the problem of contextual suggestion. We use the officially released collections from 2012 to 2014, and denote them as *CS2012*, *CS2013* and *CS2014* respectively. Each collection consists of a set of example suggestions and user profiles. User profile includes the ratings for each suggestion given by each user. The information provided about each example suggestion includes its name, a short description and the URL to its webpage. To gather the opinions for each suggestion, we

---

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

**Table 1** Statistics of the three TREC collections

Collection	No. of users	No. of suggestions	The range of ratings
CS2012	34	49	[−1,1]
CS2013	562	50	[0,4]
CS2014	299	100	[0,4]

crawl the ratings and text reviews of the suggestions from Yelp. The statistics of these three TREC collections are summarized in Table 1.

- *Yelp data set*<sup>3</sup> In the TREC collections, all users rated the same number of suggestions, which might not be the case in reality, e.g., sparse observations of users' preferences. To assess the proposed methods in a more realistic setting, we construct another evaluation data set based on Yelp reviews. Specifically, we randomly picked 100 Yelp users, and crawled the information about suggestions they had rated as example suggestions in one month period (from January 15, 2013 to February 14, 2013). Note that, for each suggestion, we have its name but do not have the short description as in the TREC collection. The total number of crawled suggestions is 13,880. All the opinions (i.e., ratings and text reviews) about each suggestion are also crawled. The users ratings are in the range of [1, 5].

These two evaluation data sets have distinct characteristics. In the TREC collections, there is a fixed set of example suggestions, and all the users provide their ratings on those suggestions. On the contrary, in the Yelp collection, different users would rate different sets of suggestions, where the overlapped suggestions are small and the number of rated suggestions per user also varies. The average number of rated suggestions per user is around 200.

## 6.2 Experiments on candidate suggestion ranking

### 6.2.1 Experiment design

In all the collections, for each user, we need to split the suggestions that rated by this user into development set and test set. The suggestions in the development set are used to construct user profile while those in the test set are used as candidate suggestions that need to be ranked. For each user, we randomly select 50 % of the suggestions from each category at each rating level to build the user profile, and use the remaining ones as the test set. We will discuss the impact of the size of development set for user profile construction in Sect. 6.2.3.

As discussed in Sect. 6.1, user rating values in different evaluation collections are different. We need to map them into either POS (i.e, positive) or NEG (i.e., negative) as described in Eq. 1. In the *CS2012* data set, the rating of 1 is mapped to POS and the ratings of −1,0 are mapped to NEG. In the *CS2013* and *CS2014* data sets, the ratings higher than 2 are mapped to POS while those lower than 2 are mapped to NEG. In the *Yelp* data set, the ratings higher than 3 are mapped to POS while those lower than 3 are mapped to NEG. Note that the reviews assigned with the middle rating are not included in the mapping

<sup>3</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).

because it is difficult to directly classify them into positive or negative opinions without looking at the text reviews.

The evaluation measures for candidate suggestion rankings are P@5 (precision at top 5 results) and ERR@20 (expected reciprocal rank at top 20 results) (Chapelle et al. 2009). P@5 is the official measure used in the TREC contextual suggestion track. Since the relevance judgement of a candidate suggestion is graded and P@5 cannot capture the graded relevance, we use ERR@20 as an additional measure.

To evaluate the effectiveness of the proposed opinion-based user profile, we compare it with two baseline methods using different types of information to build the user profiles (Yang and Fang 2012). The first one is to use text description of candidate places to build user profile and use the website content of a candidate suggestion as the candidate representation. The similarity scores between user profiles and candidate suggestions are computed using F2EXP as shown in Eq. 4. The second baseline is to use category information to build user profile and candidate suggestion profile. The similarity between a profile  $a$  and a candidate suggestion  $b$  is then computed as follows:

$$SIM_C(a, b) = \sum_{c_i \in C(a)} \sum_{c_j \in C(b)} \frac{|c_i \cap c_j|}{\max(|c_i|, |c_j|)} \times \frac{1}{|C(a)| \times |C(b)|} \quad (5)$$

where  $C(a)$  is a set of hierarchical categories for  $a$  because suggestion  $a$  may have multiple categories. Each category  $c(a)$  includes all the hierarchical information and can be represented as a set of category names, e.g., [steak house, american restaurant, restaurant].  $|c_i \cap c_j|$  is the number of common categories between  $c_i$  and  $c_j$ .

Unlike the opinion-based method where there are two representations (i.e., positive and negative) for both user profile and candidate suggestions, the two baseline methods have two representation for the user profile (i.e., the positive and negative ones) but only one for the candidate suggestions. Thus, the relevance score is computed based on two similarity functions:  $SIM(U_+(U), REP(CS))$  and  $SIM(U_-(U), REP(CS))$ . And these two functions can be combined using either linear interpolation or learning to rank methods as described in Sect. 4.

### 6.2.2 Results of candidate suggestion ranking

We first conduct experiments to evaluate the proposed opinion-based method as well as two baseline methods when using linear interpolation. The results of using 5-fold cross validation are shown in Table 2. The Yelp data set does not have description for each suggestion to build the user profile, so the description-based method is not applicable for this data set.

It is clear that opinion-based methods consistently outperform the two baseline methods over both measures and all the collections. These results show that it is more effective to model user preferences using the opinions about the suggestions than using the description or the categories of the suggestions. In particular, the improvement is larger on the Yelp data collection. This indicates that the opinion-based methods can capture the user preferences in a more general way. Moreover, the evaluation results of all the opinion-based methods are quite similar; among them, NR seems to be the most stable one.

There are four parameters in the linear interpolation methods as described in Sect. 4. We find that the optimal parameter setting is as follows:  $\alpha = 1.0$ ,  $\beta = 0.0$ ,  $\gamma = 0.9$ ,  $\eta = 0.1$ , which indicates both positive and negative user profiles are important. It verifies our

**Table 2** 5-fold cross validation results using linear interpolation method

Collections	Methods	ERR@20	P@5
CS2012	Category	0.79	0.65
	Description	0.70	0.51
	FR	0.80* <sup>†</sup>	0.68* <sup>†</sup>
	SR	0.80* <sup>†</sup>	0.66* <sup>†</sup>
	NR	0.81* <sup>†</sup>	0.66* <sup>†</sup>
	RS	0.81* <sup>†</sup>	0.67* <sup>†</sup>
CS2013	Category	0.66	0.68
	Description	0.65	0.65
	FR	0.72* <sup>†</sup>	0.70* <sup>†</sup>
	SR	0.71* <sup>†</sup>	0.69* <sup>†</sup>
	NR	0.71* <sup>†</sup>	0.70* <sup>†</sup>
	RS	0.69* <sup>†</sup>	0.68* <sup>†</sup>
CS2014	category	0.72	0.74
	description	0.71	0.74
	FR	0.73* <sup>†</sup>	0.76* <sup>†</sup>
	SR	0.71	0.77* <sup>†</sup>
	NR	0.75* <sup>†</sup>	0.78* <sup>†</sup>
	RS	0.75* <sup>†</sup>	0.75* <sup>†</sup>
Yelp	Category	0.70	0.73
	Description	–	–
	FR	0.81*	0.90*
	SR	0.81*	0.90*
	NR	0.81*	0.91*
	RS	0.81*	0.90*

\* (or <sup>†</sup>) indicates the improvement over the category-based (or description-based) method is statistically significant

hypothesis that it is necessary to capture both what a user likes and what a user dislikes in contextual suggestion. Furthermore, we can find that the positive candidate suggestion representation is more useful than the negative one.

Table 3 shows the performance of learning-to-rank methods. All the models are trained on 60 % of the data, validated on 20 % of the data, and then tested on the remaining data. This process is repeated 5 times and the average performance is reported. We can see that the opinion-based user profiling is still consistently better than the description or category-based methods. Among the three learning-to-rank methods, LMART and MART performed much better than the linear regression methods, and MART was the best. Among different representations, the performance is still similar, and NR remains to be a reasonable choice.

Based on the results of these two tables, it seems that the best strategy is to use NR for opinion-based representation and use MART to combine the similarities. In fact, another advantage of using MART is the possibility of incorporating more features. We leave this as our future work.

### 6.2.3 In-depth analysis

We first conduct experiments to analyze how the size of development set used to build the user profile affects the performance of these methods. In the previous experiments, for each

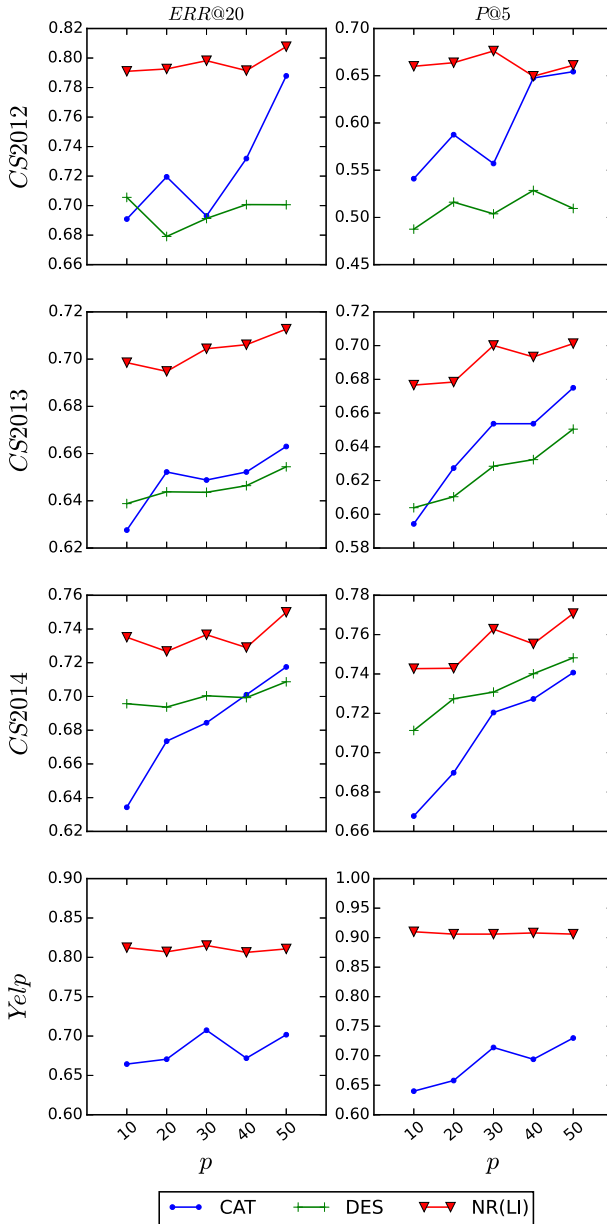
**Table 3** Performance of learning to rank methods

Collection	Feature	ERR@20			P@5		
		LR	LMART	MART	LR	LMART	MART
CS2012	Category	0.76	0.72	0.76	0.65	0.56	0.66
	Description	0.68	0.64	0.66	0.48	0.55	0.56
	FR	0.66	0.73* <sup>†</sup>	0.80* <sup>†</sup>	0.52 <sup>†</sup>	0.63* <sup>†</sup>	0.64* <sup>†</sup>
	SR	0.64	0.75* <sup>†</sup>	0.73 <sup>†</sup>	0.47	0.63* <sup>†</sup>	0.56
	NR	0.64	0.74* <sup>†</sup>	0.75 <sup>†</sup>	0.47	0.63* <sup>†</sup>	0.61 <sup>†</sup>
	RS	0.61	0.80* <sup>†</sup>	0.76* <sup>†</sup>	0.45	0.67* <sup>†</sup>	0.63 <sup>†</sup>
CS2013	Category	0.65	0.68	0.66	0.70	0.67	0.68
	Description	0.65	0.66	0.65	0.62	0.62	0.65
	FR	0.65 <sup>†</sup>	0.69* <sup>†</sup>	0.72* <sup>†</sup>	0.63 <sup>†</sup>	0.68* <sup>†</sup>	0.70* <sup>†</sup>
	SR	0.65 <sup>†</sup>	0.69* <sup>†</sup>	0.71* <sup>†</sup>	0.57	0.68* <sup>†</sup>	0.69* <sup>†</sup>
	NR	0.65 <sup>†</sup>	0.70* <sup>†</sup>	0.71* <sup>†</sup>	0.64 <sup>†</sup>	0.68* <sup>†</sup>	0.70* <sup>†</sup>
	RS	0.65 <sup>†</sup>	0.69* <sup>†</sup>	0.71* <sup>†</sup>	0.59	0.70* <sup>†</sup>	0.70* <sup>†</sup>
CS2014	Category	0.70	0.69	0.71	0.75	0.70	0.75
	Description	0.71	0.68	0.71	0.74	0.70	0.75
	FR	0.67	0.75* <sup>†</sup>	0.76* <sup>†</sup>	0.66	0.76* <sup>†</sup>	0.79* <sup>†</sup>
	SR	0.62	0.70* <sup>†</sup>	0.75* <sup>†</sup>	0.60	0.72* <sup>†</sup>	0.78* <sup>†</sup>
	NR	0.67	0.73* <sup>†</sup>	0.75* <sup>†</sup>	0.68	0.77* <sup>†</sup>	0.79* <sup>†</sup>
	RS	0.66	0.73* <sup>†</sup>	0.74* <sup>†</sup>	0.63	0.76* <sup>†</sup>	0.79* <sup>†</sup>
Yelp	Category	0.69	0.67	0.68	0.72	0.56	0.72
	FR	0.78*	0.77*	0.78*	0.84*	0.76*	0.89*
	SR	0.77*	0.80*	0.79*	0.85*	0.81*	0.93*
	NR	0.80*	0.76*	0.80*	0.85*	0.77*	0.93*
	RS	0.79*	0.76*	0.79*	0.85*	0.73*	0.92*

\* (or <sup>†</sup>) indicates the improvement over the category-based (or description-based) method is statistically significant

user, we used 50 % of the suggestions rated by the user to build user profiles. It is necessary to verify how the performance changes when fewer suggestions are used. The results are shown in Fig. 4. The X axis indicates the percentage of suggestions used to build the profile, and the Y axis corresponds to the ranking performance. It is clear that the performance of the opinion-based method (i.e., NR) is more robust with respect to the quality of the user profile. Even when we use fewer number of suggestions to build the profile, the performance remains robust.

Previous results show that NR seems to be more robust and effective than the other profile representations. Our result analysis suggests that the better performance may be related to the fact that the NR-based profiles contain fewer noisy terms. Here, we use an example pair i.e., user (uid:918) and candidate suggestion (id:107), to illustrate it. Table 4 shows the most frequent terms in the positive user profiles and the positive representation of the candidate suggestion. We can see that the candidate is about a place that selling “breakfast items” while the user seems to like “beers” and “chicken wings”. Comparing these different profiles, it is clear that the profiles generated by NR contain fewer noisy



**Fig. 4** The performance of using less data to build user profile

terms than others. When computing the similarity between the user profile and candidate suggestions, these noisy terms could mistakenly boost the ranking of the candidate suggestion. This effect has been shown in Table 5. We use KL-Divergence to measure the difference between the user profile from the candidate representation. It is clear that NR is able to capture difference between the user profile and the candidate suggestion and rank



**Table 4** Top frequent terms in different user profiles (id:918) and positive candidate profile (id:107)

Positive user profiles	
NR	Place, burg, time, beer, food, chicago, wing, pie, art, chicken, kuma, view, bar, wait, day, drink, people, friend, table, hour, thing, cheese, sauce, night, fry
FR	Burg, place, go, good, get, wait, time, great, beer, like, just, one, food, love, chicago, really, best, kuma, order, friend, will, also, back, bar, wing
SR	Order, go, burg, beer, worth, wing, will, went, well, way, want, wait, visit, view, two, try, time, though, think, take, table, sure, still, something, service
RS	Great, good, place, best, burg, amaze, time, favorite, beer, pie, chicago, food, art, view, first, nice, ever, delicious, beautiful, fan, awesome, worth, wait, friend, free
Positive Candidate Profile	
Name	Little Goat
Description	Upscale diner next to the Little Goat Bakery serving breakfast items, sandwiches, burgers & more
	Goat, wait, little, good, food, great, order, place, like, dine, time, go, menu, love, just, try, back, friend, get, really, delicious, also, one, breakfast, sandwich, cheese, got, table, pork, service, will, pancake, come, serve, coffee, well, can, amaze, definite, bread

**Table 5** KL divergence between positive user profile (id:918) and positive candidate profile (id:107)

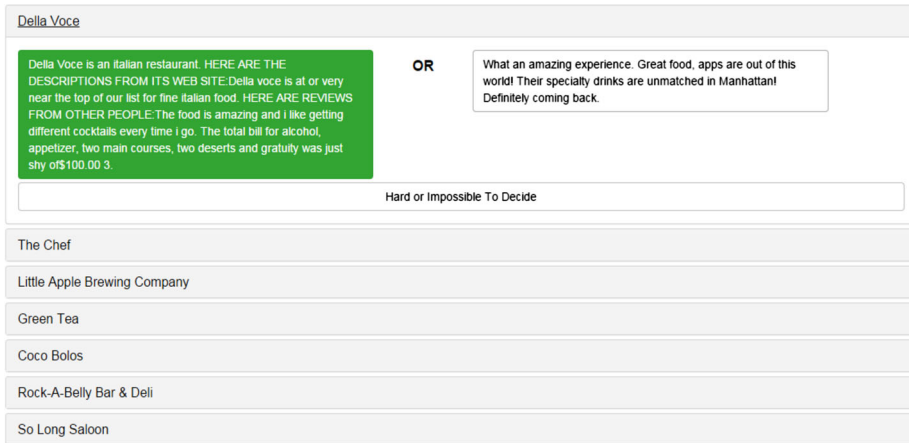
Representations	KL Div.	Ranking
NR	1.54	7
FR	0.61	2
SR	1.40	2
RS	0.95	5

the suggestion at the seventh place. On the other hand, the other representations are more similar to the candidate suggestion and incorrectly rank it at a higher place.

### 6.3 Experiments on summary generation

We conduct two sets of experiments to evaluate the proposed structured summary generation method.

We first evaluate the quality of the summaries generated by the proposed method. The baseline method is the snippet generation method developed by Yelp, and this method was used in one of the top ranked TREC runs (Dean-Hall et al. 2014). To compare the results of the two methods, we develop an annotation interface as shown in Fig. 5. There are 2109 unique suggestions from the TREC 2013 and 2014 contextual suggestion tracks, and we generate the summary for each of them using the two methods. For each suggestion, the annotation system would present the summary generated by the two methods, and annotators are expected to read the results and decide which one is better or choose “Hard or Impossible to Decide”. Two annotators are hired for this task, and they are assigned to



**Fig. 5** Screen shot of the web-based annotation system to compare two summary generation methods

judge 1300 and 1209 suggestions respectively. There are suggestions judged by both assessors so that we can see whether judgements between the two assessors are consistent.

The comparison results are shown in Table 6. Among the overlapped suggestions, both annotators think that our method performs better than the baseline method for over 70 % of the suggestions. Similar observations can be made for the non-overlapped suggestion set as well. Thus, it is clear that our structured summary generation method is more effective than the state of the art baseline method.

Since each structured summary contains multiple components, we also conduct experiments to evaluate the effectiveness for each component. Note that the last component is personalized and it is trivial to evaluate its effectiveness, so we focus on evaluating the first three components, i.e, opening, official introduction and review. We recruit three annotators (two of whom are the same ones as in the previous task) to assess the quality of the structured summaries. Following the same judgement strategy used by TREC, there are 5 rating levels, and 0, 1, 2 are mapped to non-relevant and 3, 4 are mapped to relevant. The interface of the annotation system is shown in Fig. 6. Again, there are 2109 suggestions, and we split the job among three annotators. There are 200 suggestions assessed by all the

**Table 6** Comparison of results summarization methods

	Annotator#1 (%)	Annotator#2 (%)
<i>Overlapped suggestions</i>		
Our method is better than the baseline	71	86
Our method is worse than the baseline	20	11
Hard or impossible to decide	9	4
<i>Non-overlapped suggestions</i>		
Our method is better than the baseline	78	68
Our method is worse than the baseline	15	32
Hard or Impossible to decide	7	0

The screenshot shows a web-based annotation system for evaluating components of a restaurant review. The system displays four components, each with a checkmark, a description, a star rating, and a quality label.

- Opening:** Della Voce is an Italian restaurant. Rating: 5 stars (Very Good).
- Intro:** HERE ARE THE DESCRIPTIONS FROM ITS WEB SITE Della voce is at or very near the top of our list for fine Italian food. Rating: 3 stars (Poor).
- Review:** HERE ARE REVIEWS FROM OTHER PEOPLE. The food is amazing and I like getting different cocktails every time I go. The total bill for alcohol, appetizer, two main courses, two deserts and gratuity was just shy of \$100.00. Rating: 4 stars (Good).
- Whole:** Della Voce is an Italian restaurant. HERE ARE THE DESCRIPTIONS FROM ITS WEB SITE. Della voce is at or very near the top of our list for fine Italian food. HERE ARE REVIEWS FROM OTHER PEOPLE. The food is amazing and I like getting different cocktails every time I go. The total bill for alcohol, appetizer, two main courses, two deserts and gratuity was just shy of \$100.00. Rating: 4 stars (Good).

**Fig. 6** Screen shot of the web-based annotation system to evaluate the effectiveness of components

**Table 7** Evaluation results on the overlapped suggestions (measured by accuracy)

Components	Annotator #1	Annotator #2	Annotator #3
Opening	0.98	0.81	0.80
“Official” Intro	0.75	0.53	0.78
Review	0.87	0.95	0.99

**Table 8** Evaluation results on all the suggestions (measured by accuracy)

Components	CS2013	CS2014
Opening	0.99	0.83
“Official” Intro	0.56	0.47
Review	0.69	0.77

three assessors to measure the agreement. The results are evaluated with accuracy, i.e., the number of relevant summaries divided by the number of summaries.

Table 7 shows the accuracy of each section for the overlapped suggestions. It is clear that all sections have high accuracy. Among them, it seems that the official introduction are less relevant than the other two components. We also measure the agreement among the three assessors, and the agreement is around 0.5 for the official introduction, 0.7 for the opening, and 0.6 for the review component. Furthermore, Table 8 shows the accuracy of each component for all suggestions including the ones shared among annotators. If a suggestion is from the overlapped set, i.e., having more than one annotation, the relevance status of a component is determined by the majority vote. Since all the suggestions are from the pool of either TREC 2013 CS track or TREC 2014 CS track, we report the accuracy for each collection separately. The observation here is similar to what we observed in the overlapped set. It is clear that both opening and review components are useful and more relevant.

## 7 Related work

### 7.1 TREC contextual suggestion track

The problem of contextual suggestion was first introduced at TREC in 2012, and the track has been running in the past three years Dean-Hall et al. (2013), Dean-Hall et al. (2012). Although the details of the track varied, the task remains the same. Given a user's preferences on a set of example suggestions and a context, track participants are expected to return a ranked list of new suggestions that are likely to satisfy both the user preferences (based on their preferences on the example suggestions) as well as the contexts such as geotemporal locations. Each example suggestion includes a title, description and an associated URL. For each user, we know their preferences on part or all of the example suggestions.

Most TREC participants retrieved candidate suggestions from various online services such as Google Place or Yelp based on the geographical context and then use some heuristics, e.g. nightclub will not be shown if the temporal context is in the morning, to filter out the suggestions that do not match the temporal contexts (Dean-Hall et al. 2013, 2012). After that, the task is to retrieve useful suggestions based on user preferences. Most participants formulated the task as a content-based recommendation problem (Hubert and Cabanac 2012; Jiang and He 2013; Li and Alonso 2014; Li et al. 2014; McCreadie et al. 2014; Rao and Carterette 2012; Roy et al. 2013; Xu and Callan 2014; Yang and Fang 2012, 2014; Yates et al. 2012). A common strategy adopted by top-ranked participants of TREC is to estimate a user profile based on the example suggestions and then rank candidate suggestions based on their similarities to the user profile. The basic assumption is that a user would prefer suggestions that are similar to those example suggestions liked by the user. Various types of information about the suggestions have been used to estimate user profiles which include the description of the places (Hubert and Cabanac 2012; Jiang and He 2013; Yang and Fang 2012, the categories of the places Koolen et al. 2013; Li and Alonso 2014; Li et al. 2014; McCreadie et al. 2014; Yates et al. 2012, and the web sites of the places Hubert and Cabanac (2012); Jiang and He 2013; Yang and Fang 2012).

Specifically, many studies used terms from the description of the places or the web pages of the example suggestions to construct user profiles, and then various similarity measures are used to rank the candidates (Hubert and Cabanac 2012; Jiang and He 2013; Yang and Fang 2012). A few studies also explored the use of category information for user profiling and candidate ranking. For example, Li and Alonso (2014) utilized the accumulative category scores to model both user and candidate profiles, and then use the full range cosine similarity between the two profiles for candidate ranking. Li et al. (2014) leveraged how likely each popular category is liked/disliked by users to construct user profiles, and the candidate ranking is to favor suggestions from a user's favorite categories. McCreadie et al. (2014) proposed to rank the candidates by comparing two trees of finer-grained categories between user profile and candidate profile using a tree-matching technique. Diversification is then applied so that the categories of top ranked candidates are normalized. Yates et al. (2012) proposed to recommend the candidates which are proportional to the number of example suggestions in each category. Koolen et al. (2013) applied a similar method with a major modification of retrieving the category information from Wikitravel.<sup>4</sup>

<sup>4</sup> <http://www.wikitravel.org/>.

However, none of other groups has tried to leverage the reviews about these places to estimate the user profile as what we propose in this paper. As we mentioned earlier, using either descriptions or categories can not precisely capture what a user likes or dislikes. However, online reviews offer rich information about user opinions and should be leveraged in user profiling. To the best of our knowledge, we are the first ones who incorporate opinions as user profiles in pursuing better solution for contextual suggestion.

## 7.2 Recommendation systems

The problem of contextual suggestion is also similar to collaborative filtering (Su and Khoshgoftaar 2009). Collaborative filtering assumes that similar users would share similar ratings, and focuses on predicting the user rating based on such an assumption. It often requires a large number of past user preferences to be more accurate and sometimes it may suffer from data sparsity problem which is known as the cold start problem (Schein et al. 2002). In order to solve the data sparsity problem, reviews were incorporated to improve the performance. Hariri et al. (2011) inferred the context or the intent of the trip by analyzing reviews. In particular, they used latent Dirichlet Allocation to identify the topics from the reviews, and the final ranking scores are generated based on both the context scores as well as the scores generated by traditional collaborative filtering methods. Jakob et al. (2009) proposed to cluster the features and then apply natural language processing techniques to identify the polarity of the opinions. A few studies also focused on leveraging Location Based Social Network to solve the data sparsity problem. Noulas et al. (2012) applied random walk based on latent space models and computed a variety of similarity criteria with venue's visit frequencies on the location based social network. Bao et al. (2012) proposed to first constructing a weighted category hierarchy and then identify local experts for each category. The local experts are then matched to a given user and the score of the candidate is inferred based on the opinions of the local experts.

Our work is also related to other studies that utilized reviews to improve the performance of recommendation systems (Hariri et al. 2011; Levi et al. 2012; Qumsiyeh and Ng 2012; Raghavan et al. 2012; San Pedro et al. 2012). Raghavan et al. (2012) proposed to use the helpfulness, features from the text reviews and the meta-data (average rating, average length of text reviews and etc.) of the opinions to train a regression model in order to generate a quality score for each opinion. The quality score is then incorporated into the probabilistic matrix factorization as an inverse factor which affects the variance of the prediction from the mean of the factor model. Levi et al. (2012) extended this study and analyzed the review texts to get the intent, features and the ratings for each feature. Qumsiyeh and Ng (2012) explored the aspects in the reviews and computed the probability of each genres (categories) in each rating level. Their work is limited to the applications in multimedia domains, and the genres of each type of media is pre-defined.

Our work is different from these previous studies in the following aspects. First, our focus is to directly use reviews to model user profile while previous studies mainly used reviews to predict the rating quality or the user intent. Second, existing studies on collaborative filtering were often evaluated on only specific applications, e.g., movies, hotels, and it is unclear how those methods could be generalized to other domains. In contrast, our proposed method is not limited to any specific domains and can be applied to a more general problem set up.

### 7.3 Text summarization

The summary generation of our work is related to automatic text summarization. Automatic text summarization has been well studied for traditional documents such as scientific documents and news articles (Radev et al. 2002). In particular, previous work has studied various problems in this area including extractive summarization, abstractive summarization, single-document summarization and multiple-document summarization (Das and Martins 2007). More recently, there have been effort on opinion mining and summarization (Baccianella et al. 2010; Chen and Zimbra 2010; Dey and Haque 2009; Ding and Liu 2007; Esuli 2008; Knight and Marcu 2002; Mani 2001; Mani and Maybury 1999; Pak and Paroubek 2010; Pang and Lee 2004, 2008). Most of them involve in the finer partition of the reviews and polarity judging of each partition. Common strategies include part-of-speech analysis, negation identification and etc. Unlike the previous effort, we focus on generating a *personalized* summary for a suggestion. Since the information about the suggestion is scattered in many places, including description, web sites and reviews, the summarization needs to synthesize the information from these heterogeneous information sources. Instead of extracting the information from a single source, we try to leverage one information source to guide the extractive summarization process in other sources and then assemble all the extracted summaries together into a *structural* way. Another main difference of our work from previous studies is to utilize the user profile to generate personalized summaries.

## 8 Conclusions and future work

One of the important challenging in mobile search is to return satisfying results based on not only user preference history but also contextual information. This paper focuses on user profiling for contextual suggestion. In particular, we propose to use opinions for both candidate suggestion ranking as well as suggestion summary generation. For candidate suggestion ranking task, opinions are used to build the enriched user profile and the candidate suggestion profile. Several opinion representations including the full review text and part of the review text have been explored to model the user profile. Various ranking methods including linear interpolation and learning to rank have been investigated to compute the ranking scores. The effectiveness of our method has been tested by comparing the results with category and description baselines on standard TREC collections and a self-crawled Yelp collection. The results showed that our method significantly outperforms the baselines in terms of both  $P@5$  and  $ERR@20$ . The more detailed analysis showed that using NR as the representation of the opinion in general performs better than other opinion representations. Although there is no obvious difference between the optimal performance of linear interpolation and learning to rank, MART performs better than other learning to rank methods—Lambda MART and Linear Regression. We further did more in-depth analysis on how size of the user profile can affect the performance. The results showed that using as few as 10 % of the user preferences to build the user profile still leads to promising performance. Furthermore, we also propose to construct a structured summary by leveraging information from multiple resources. Experiment results show that the proposed method is more effective than the baseline method, and some components in the summary are more useful than the others.

There are many interesting directions that we plan to pursue in the future. First, it would be interesting to evaluate the proposed method in the personalized local search problem (Lv et al. 2012). Second, we only focus on the user modeling in this paper, and plan to study how to incorporate other context-related factors such as distances and recency into the ranking process. Finally, it would be interesting to explore a more personalized summary generation method.

**Acknowledgments** This material is based upon work supported by the National Science Foundation under Grant Number IIS-1423002 and the National Basic Research Program of China (973 Program) under Grant 2013CB336500. We thank the reviewers for their useful comments.

## References

- Allan, J., Croft, B., Moffat, A., & Sanderson, M. (2012). Frontiers, challenges, and opportunities for information retrieval: Report from swirl 2012 the second strategic workshop on information retrieval in lorne. *SIGIR Forum*, 46(1), 2–32.
- Baccianella, S., Esuli, A., & Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the seventh international conference on language resources and evaluation (LREC'10)*, Valletta, Malta, May 2010.
- Bao, J., Zheng, Y., & Mokbel, M. F. (2012). Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th international conference on advances in geographic information systems, SIGSPATIAL '12, New York, NY, USA, 2012* (pp. 199–208). ACM.
- Bellogín, A., Gebremeskel, G. G., He, J., Lin, J., & Said, A. (2013). CWI and TU delft notebook TREC 2013: Contextual suggestion, federated web search, KBA, and web tracks. In *Proceedings of TREC'13*.
- Burges, C. J. C. (2010). *From RankNet to LambdaRank to LambdaMART: An overview*. Technical report, Microsoft Research.
- Chapelle, O., Metzler, D., Zhang, Y., & Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on information and knowledge management, CIKM '09, New York, NY, USA, 2009* (pp. 621–630). ACM.
- Chen, H., & Zimbra, D. (2010). Ai and opinion mining. *IEEE Intelligent Systems*, 25(3), 74–80.
- Das, D., & Martins, A. F. T. (2007). A survey on automatic text summarization. In *Literature Survey for the Language and Statistics II course at Carnegie Mellon University*. <http://www.cs.cmu.edu/~nasmith/LS2/das-martins.07.pdf>.
- Dean-Hall, A., Clarke, C., Kamps, J., Thomas, P., Simone, N., & Voorhees, E. (2013). Overview of the TREC 2013 contextual suggestion track. In *Proceedings of TREC'13*.
- Dean-Hall, A., Clarke, C., Kamps, J., Thomas, P., Simone, N., & Voorhees, E. (2014). Overview of the TREC 2014 contextual suggestion track. In *Proceedings of TREC'14*.
- Dean-Hall, A., Clarke, C., Kamps, J., Thomas, P., & Voorhees, E. (2012). Overview of the TREC 2012 contextual suggestion track. In *Proceedings of TREC'12*.
- Dey, L., & Haque, S. M. (2009). Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition (IJ DAR)*, 12(3), 205–226.
- Ding, X., & Liu, B. (2007). The utility of linguistic rules in opinion mining. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 811–812). ACM.
- Esuli, A. (2008). Automatic generation of lexical resources for opinion mining: Models, algorithms and applications. *SIGIR Forum*, 42(2), 105–106.
- Fang, H., & Zhai, C. (2005). 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, New York, NY, USA* (pp. 480–487).
- Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29, 1189–1232.
- Ganesan, K., Zhai, C., & Han, J. (2010). Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics, COLING '10* (pp. 340–348).

- Hariri, N., Mobasher, B., Burke, R., & Zheng, Y. (2011). Context-aware recommendation based on review mining. In *Proceedings of the 9th workshop on intelligent techniques for web personalization and recommender systems*.
- Hubert, G., & Cabanac, G. (2012). IRIT at TREC 2012 contextual suggestion track. In *Proceedings of TREC'12*.
- Jakob, N., Weber, S. H., Müller, M. C., & Gurevych, I. (2009). Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion, TSA '09, New York, NY, USA, 2009* (pp. 57–64). ACM.
- Jiang, M., & He, D. (2013). PITT at TREC 2013 contextual suggestion track. In *Proceedings of TREC'13*.
- Knights, K., & Marcu, D. (2002). Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1), 91–107.
- Koolen, M., Huurdeman, H., & Kamps, J. (2013). University of Amsterdam at the TREC 2013 contextual suggestion track: Learning user preferences from wikitravel categories. In *Proceedings of TREC'13*.
- Levi, A., Mokryn, O., Diot, C., & Taft, N. (2012). Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of the RecSys'12*.
- Li, H., & Alonso, R. (2014). User modeling for contextual suggestion. In *Proceedings of TREC'14*.
- Li, H., Yang, Z., Lai, Y., Duan, L., & Fan, K. (2014). BJUT at TREC 2014 contextual suggestion track: Hybrid recommendation based on open-web information. In *Proceedings of TREC'14*.
- Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 225–331.
- Lv, Y., Lymberopoulos, D., & Wu, Q. (2012). An exploration of ranking heuristics in mobile local search. In *Proceedings of the SIGIR'12*.
- Macdonald, C., Santos, R. L., & Ounis, I. (2013). The whens and hows of learning to rank for web search. *Information Retrieval*, 16(5), 584–628.
- Mani, I. (2001). *Automatic summarization* (Vol. 3). Amsterdam: John Benjamins.
- Mani, I., & Maybury, M. T. (1999). *Advances in automatic text summarization*. Cambridge, MA: MIT Press.
- McCreadie, R., Deveaud, R., Albakour, M.-D., Mackie, S., Limsopatham, N., Macdonald, C., Ounis, I., Thonet, T., & Taner, B. (2014). University of Glasgow at TREC 2014: Experiments with terrier in contextual suggestion, temporal summarisation and web tracks. In *Proceedings of TREC'14*.
- Noulas, A., Scellato, S., Lathia, N., & Mascolo, C. (2012). A random walk around the city: New venue recommendation in location-based social networks. In *Proceedings of the 2012 ASE/IEEE international conference on social computing and 2012 ASE/IEEE international conference on privacy, security, risk and trust, SOCIALCOM-PASSAT '12, Washington, DC, USA, 2012* (pp. 144–153). IEEE Computer Society.
- Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the seventh conference on international language resources and evaluation (LREC'10), Valletta, Malta, May 2010*.
- Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on association for computational linguistics, ACL '04, Stroudsburg, PA, USA*.
- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2), 1–135.
- Qumsiyeh, R., & Ng, Y.-K. (2012). Predicting the ratings of multimedia items for making personalized recommendations. In *Proceedings of SIGIR'12*.
- Radev, D., Jing, H., & Budzikowska, M. (2002). Introduction to the special issue on summarization. *Computational Linguistics*, 28(4), 399.
- Raghavan, S., Gunasekar, S., & Ghosh, J. (2012). Review quality aware collaborative filtering. In *Proceedings of RecSys'12*.
- Rao, A., & Carterette, B. (2012). Udel at TREC 2012. In *Proceedings of TREC'12*.
- Roy, D., Bandyopadhyay, A., & Mitra, M. (2013). A simple context dependent suggestion system. In *Proceedings of TREC'13*.
- San Pedro, J., Yeh, T., & Oliver, N. (2012). Leveraging user comments for aesthetic aware image search reranking. In *Proceedings of WWW'12*.
- Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '02, New York, NY, USA, 2002* (pp. 253–260). ACM.
- Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, 421425. doi:10.1155/2009/421425.



- Xu, D., & Callan, J. (2014). Modelling psychological needs for user-dependent contextual suggestion. In *Proceedings of TREC'14*.
- Yang, P., & Fang, H. (2012). An exploration of ranking-based strategy for contextual suggestion. In *Proceedings of TREC'12*.
- Yang, P., & Fang, H. (2013). An opinion-aware approach to contextual suggestion. In *Proceedings of TREC'13*.
- Yang, P., & Fang, H. (2013). Opinion-based user profile modeling for contextual suggestions. In *Proceedings of the 2013 conference on the theory of information retrieval, ICTIR '13, New York, NY, USA, 2013* (pp. 18:80–18:83). ACM.
- Yang, P., & Fang, H. (2014). Exploration of opinion-aware approach to contextual suggestion. In *Proceedings of TREC'14*.
- Yates, A., DeBoer, D., Yang, H., Goharian, N., Kunath, S., & Frieder, O. (2012). (Not too) personalized learning to rank for contextual suggestion. In *Proceedings of TREC'12*.