

# User Model Enrichment for Venue Recommendation

Mohammad Aliannejadi, Ida Mele, and Fabio Crestani

Faculty of Informatics, Università della Svizzera Italiana, Lugano, Switzerland  
{mohammad.alian.nejadi, ida.mele, fabio.crestani}@usi.ch

**Abstract.** An important task in recommender systems is suggesting relevant venues in a city to a user. These suggestions are usually created by exploiting the user’s history of preferences, which are, for example, collected in previously visited cities. In this paper, we first introduce a user model based on venues’ categories and their descriptive keywords extracted from Foursquare tips. Then, we propose an enriched user model which leverages the users’ reviews from Yelp. Our participation in the TREC 2015 Contextual Suggestion track, confirmed that our model outperforms other approaches by a significant margin.

## 1 Introduction

Recent years have witnessed an increasing use of location-based social networks (LBSNs) such as Yelp, TripAdvisor, and Foursquare. These social networks collect valuable information about users’ mobility records, which often consist of their check-in data and may also include users’ ratings and reviews. Therefore, being able to recommend personalized venues to users plays a key role in satisfying the user needs on such social networks.

One of the challenges in recommending venues is to model the user based on her profile (e.g., the ratings of previously visited venues). In the past, researchers proposed to make recommendations based on the similarity between the users’ preferences and the venues’ description and categories [12]. Others leveraged the opinions of users about a given place, which are, for example, extracted from the users’ online reviews [14]. We believe that these two techniques should be used together to get better recommendations.

Recent research has focused on recommending venues using collaborative-filtering technique [8, 15], where the system recommends venues based on users whose preferences are similar to those of the target user (i.e., the user who receives the recommendations). Collaborative-filtering approaches are very effective, but they suffer from the cold-start (i.e., they need to collect enough information about a user for making recommendations) and the data-sparseness problems. Furthermore, these approaches mostly rely on check-in data to learn the preferences of users, and such information is insufficient to get a complete picture of what the user likes or dislikes in a specific venue (e.g., the food, the view). In order to overcome this limitation, we model the users by applying

deeper analysis on users' past ratings as well as their reviews. In addition, following the principle of collaborative filtering, we exploit the reviews from different users with similar preferences.

In this paper we present a novel approach for suggesting venues to users, where the users are modeled based on venues' content as well as users' reviews. For the former we use the categories of the venues enriched by keywords extracted from users' online reviews, which provide a more detailed description of the venue itself. Although the venue information is valuable for inferring "what type" of places a user may like or dislike, it does not give any clue on the reasons "why" a user rated as positive or negative a particular place. We need to exploit the user's opinions in order to understand what the user may have appreciated of a place. One way to obtain these opinions is mining the users' reviews and see how much they liked the venue and, more importantly, for which reasons: was it for the quality of food, for the good service, for the cozy environment, or for the location? In cases where we lack reviews from some of the users (e.g., they have rated a venue but omitted to review it) and therefore we cannot extract opinions, we apply the collaborative-filtering principle and we use reviews from other users with similar interests and tastes. Our intuition is that a user's opinion regarding an attraction could be learned based on the opinions of others who expressed the same or similar rating for the same venue. To do this we exploit information from multiple sources and combine them to gain better performance. We report the results of our participation in the TREC Contextual Suggestion Track 2015 which show how our model outperforms all the other runs by a significant margin and is placed as the first run in the track.

The remainder of the paper is organized as follows. Section 2 reviews related work. Then, we present our methodology in Section 3. Section 4 describes our experiments. Finally, Section 5 is a short conclusion and description of future work.

## 2 Related Work

Recommendation systems help users to find interesting items in large collections. These systems can be employed for recommending products (e.g., books, songs), information (e.g., news and blog articles), or venues (e.g., restaurants, pubs). In recent years, due to the availability of Internet access on mobile devices, there has been a large interest in venue recommendation and contextual suggestion [6] given that the context would be easily provided by the mobile device.

Recommendation algorithms can be divided into four categories: content-based approaches, rating-based collaborative filtering, preference-based product ranking, and review-based approaches [3].

Content-based approaches build user and item profiles based on items' contents (i.e., description, meta-data, keywords) and measure the similarity between the profiles. In [12], the authors applied Part-of-Speech (POS) tagging to the venues' descriptions in order to get the most informative terms which are then

expanded using the synonyms from WordNet<sup>1</sup>. Venues' description and categories can be helpful to infer users' preferences and, in particular, the type of places the user likes, but they do not enable understanding the reasons behind a positive or negative rate of the user for a venue, and this is considered one of the limitations of such approaches.

Rating-based collaborative filtering approaches are based on finding out common features among users' preferences and recommending venues to people with similar interests. These models are usually based on matrix factorization for dealing with huge collections of data, and they mostly use check-in data for recommending places [4, 9]. The usual assumption is that if the user goes to a place multiple times, she probably likes it, but this does not take into account the users' ratings and their reviews. In [11], the authors utilize factorization machines to leverage the feedback of the user as well as contextual information for improving the venue recommendation. These approaches usually suffer from the data-sparsity problem.

Preference-based product ranking is applied when an item (venue) can be described as a set of attributes such as price, view, staff, etc. A user can be modeled as a weighted combination of all the attributes that represent how much a particular user cares about a specific attribute and/or how it affects a user's opinion about an item (venue)[3]. Unfortunately, due to the lack of such data, such techniques cannot be applied to the venue-recommendation scenario.

Review-based approaches aim to build enhanced user profiles using their reviews. When a user writes a review about a venue, there is a wealth of information which reveals the reasons why that particular user is interested in a venue or not. Chen et al. [3] state three main reasons for which the reviews can be beneficial for a recommender system: (1) extra information that can be extracted from reviews enables a system to deal with large data sparsity problem; (2) reviews have been proven to be helpful to deal with the cold-start problem; (3) even in cases when the data is dense, they can be used to determine the quality of the ratings or to extract user's contextual information. As an example, Hariri et al. [10] tried to predict user's context from their reviews about venues by learning a Labeled Latent Dirichlet Allocation model on a dataset from TripAdvisor and using the predicted contextual information to measure the relevance of a venue to a user.

Researchers, observing the effectiveness of reviews for recommending venues [2, 7], have been motivated to model the users based on reviews. To overcome the problem that for some users there might be no reviews, Yang et al. [14] demonstrated how it is possible to get improved recommendations by modeling a user with the reviews of other users' whose tastes are similar to the ones of the target user. In particular, they modeled users by extracting positive and negative reviews to create positive and negative profiles for users and venues. The recommendation is then made by measuring and combining the similarity scores between all pairs of profiles. Inspired by their work, we also use reviews

---

<sup>1</sup> WordNet - <http://wordnet.princeton.edu>

of users with similar tastes, but instead of applying a simple similarity measure between venues and users, we use a binary classification.

In this paper, we propose a novel method for recommending venues that builds up on two models: a category-based user model to answer the question “what kind of places would a user like?” and a review-based user model, which answers the question “why would a user like a place?” Differently from other works, we combine the venues’ content and the users’ reviews to get better recommendations. Moreover, to overcome the problem of the lack of users’ reviews from which it is possible to extract users’ opinions we rely on the basic assumption of collaborative-filtering [13] and we assume that similar users tend to share similar ratings for the same venue.

### 3 User Modeling

In this section we firstly describe the user model based on the venues’ categories and keywords extracted from Foursquare’s tastes. Then, we present how to model the user with opinions extracted from reviews.

#### 3.1 Content-Based User Model

Categories of venues represent a valuable source of information that can be used to infer the types of places a user may like or dislike. Moreover, in cases where users do not provide any reviews or these are not sufficient to model their preferences, categories represent the only resource we can leverage to make venue recommendations.

To model the user’s interests using venue categories, we adopt a frequency-based approach. For simplicity we describe how we model the positive categories, while the negative-category model is built similarly. We design Algorithm 1 to calculate the category frequencies ( $cf_{pos}$ ) and build a positive category model for each user. Let  $V = \{v_1, \dots, v_M\}$  be the set of venues which were positively rated in the user’s history. Each place is associated with a set of categories,  $C(v) = \{c_1, \dots, c_z\}$ . We assume that if the user rated a venue positively, she also liked the corresponding categories. So, let  $CM_{pos}$  be the set of positively rated categories, which is made of all the categories of the venues belonging to  $V$ . We compute the frequency of these categories by counting the number of times a user rated the category positively:  $\text{count}(c_j) = \sum_{v_s \in V} \sum_{c_k \in C(v_s)} \delta(c_j, c_k)$ , where

$$\delta(c_j, c_k) = \begin{cases} 1 & \text{if } c_j = c_k \\ 0 & \text{if } c_j \neq c_k \end{cases}$$

Each category frequency in the positive (negative) category model is normalized in order to have a score between 0 and 1. Note that the users may have rated the same category with different scores depending on the venues they liked or disliked.

---

**Algorithm 1** User Positive Category Modeling

---

```
for all  $v_i \in V$  do
  for all  $c_j \in C(v_i)$  do
    if  $c_j \notin CM_{pos}$  then
       $CM_{pos} \leftarrow CM_{pos} \cup c_j$ 
       $count(c_j) = \sum_{v_s \in V} \sum_{c_k \in C(v_s)} \delta(c_j, c_k)$ 
       $N = \sum_{v_s \in V} \sum_{c_k \in C(v_s)} 1$ 
       $cf_{pos}(c_j) = count(c_j)/N$ 
    end if
  end for
end for
```

---

Given a user  $u$  and a venue  $v$ , the category-based similarity score  $S_{CM}(u, v)$  between them is calculated as follows:

$$S_{CM}(u, v) = \sum_{c_i \in C(v)} cf_{pos}(c_i) - cf_{neg}(c_i) \quad (1)$$

where  $cf_{pos}$  and  $cf_{neg}$  are respectively the positive and negative categories' frequencies.

**Foursquare's Taste Keywords.** The previous model can be enriched by using special terms extracted from users' reviews about a venue. Foursquare provides a list of keywords, also known as "tastes" to better describe a venue. As an example, 'Central Park' in 'New York City' is described by these taste terms: *picnics, biking, trails, park, scenic views*, etc. Such keywords are very informative, since they often express characteristics of a venue, and they can be considered as a complementary source of information for venue categories.

Table 1 shows all taste keywords and categories for a sample restaurant on Foursquare. As we can see, the taste keywords represent much more detailed information about the venue compared to categories. The average number of taste keywords for venues (8.73) is much higher than the average number of categories for venues (2.8). It suggests that these keywords could describe a venue in more details compared to categories.

Consequently, we consider these keywords as a complementary source of information for categories and use the same frequency-based approach to further enrich the user model. Given a user  $u$  and a place to recommend  $v$  we compute the similarity score with this category-based model enriched with Foursquare's taste terms as we did for the simple category-based model (see Equation 1) and we call it  $S_{TM}(u, v)$ .

### 3.2 Review-Based User Model

We believe that modeling a user solely based on content of venues she visited or liked is very general and would not allow to understand the specific reasons for which a user liked or disliked a place. For example, consider a user who rated two venues belonging to the same categories *Restaurant, Italian*, and *Pizza*, with

**Table 1.** A sample of taste keywords and categories for a restaurant

<b>Taste keywords</b>	pizza, lively, cozy, good for dates, authentic, casual, pasta, desserts good for a late night, family-friendly, good for groups, ravioli, lasagna, salads, wine, vodka, tagliatelle, cocktails, bruschetta
<b>Categories</b>	pizza place, italian restaurant

a positive and a negative rating, respectively. Looking only at the category and at the rates, we cannot know if the user does not like Italian restaurants and pizza places in general, or if she did not appreciate the second venue for some other reasons (e.g., food quality, service). In order to understand why the user liked or disliked a venue, we need to determine the reasons behind a positive or negative rating. This is only possible if reviews are available. In particular, analyzing the text of the reviews, we can observe that the user rated positively the first venue, because she appreciated the food and the kind service, while she did not like the second venue because of the food quality and the location.

So, to figure out for which reasons the user expressed an opinion we need to know the user’s reviews about the rated venues. Unfortunately, there is often a lack of explicit reviews from the users, so we tried to overcome this problem by using opinions expressed by other users who rated the venue with a similar score. Lacking any other information, our intuition is that a user liked/disliked a place for the same reasons that others liked/disliked that place. Although this assumption might not be perfect and might not always be valid, it provides the best way to model users in case we lack other information.

**Binary Classification.** For each user, we build a model by training a binary classifier with the positive and negative reviews of previously visited venues. We decided to use a binary classification, because we assume that a user, before planning a trip or trying a new venue, would read the online reviews of other users to have an insight on the places of interest. Suppose that the user would like to try a restaurant and, in order to decide whether it is worth to go or not, she checks the online reviews of other customers. The user may have a positive or negative idea about the restaurant depending on the ratings and comments of other people.

Subsequently, if the user rates the restaurant positively, we can assume that her judgment after reading positive reviews about the venue was positive, so she tried it and expressed an opinion similar to the other customers. An alternative to binary classification would be a regression model, but we decided not to adopt it for two reasons. First, as explained before, when users make their minds reading online reviews they have to take a binary decision: like or dislike that same place. Secondly, due to the sparsity of the dataset, a binary discrimination of venues and reviews helps our system to model users more accurately.

**Support Vector Machine.** SVM was first introduced by Cortes and Vapnik [5], and it is considered one of the most powerful supervised classifiers in machine learning. The SVM classifier model deals with binary-classification problems in which the training data is supposed to be divided into two classes using a

hyperplane which is defined by a number of support vectors. The underlying idea behind supervised learning approaches is to learn from training examples. SVM finds optimal separated hyperplanes for a binary classification problem through mapping of the input vectors into a high-dimensional feature space in a nonlinear manner. It constructs a linear model for estimating the decision function based on the support vectors. In case the training data is linearly separable, SVM results in an optimal hyperplane with maximum margin between the hyperplane and the training samples which are closest to the hyperplane, namely, the support vectors.

Our problem can be easily mapped to a binary-classification problem, as a user either likes or dislikes a venue, so we can apply successfully the SVM classifier. We separate relevant and non-relevant suggestions for each user into two classes,  $y_i \in \{-1, 1\}$ , and the number of labeled training examples is  $N$ . Therefore, the training examples are  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N), \mathbf{x} \in R^d$  where  $d$  is the number of features for each instance. The decision function without using a kernel for linearly separable training data is:

$$y_j = \text{sign}(\mathbf{w}^* \bullet \mathbf{x}_j - b),$$

where  $x_j$  is an unknown vector,  $\bullet$  represents the dot product, and  $\mathbf{w}^*$  is:

$$\mathbf{w}^* = \sum_{i=1}^r \alpha_i y_i \mathbf{x}_i,$$

where  $r$  is the number of nonzero  $\alpha$ 's.

In order to find the optimal discriminant hyperplane, one needs to find the optimal weight vector  $\mathbf{w}^*$  such that  $\|\mathbf{w}^*\|$  is the minimum. This operation can be done using Lagrangian Multipliers.

Our preliminary experiments show that among all possible kernels for SVM, linear kernel exhibits the best performance, so we choose linear kernel to train SVM classifier.

**Training the Classifier.** As we will explain in Section 4.1, for the training we used example suggestions, basically venues rated by users. In particular, positive training samples are extracted from positive reviews of positive example suggestions, while negative samples are from negative reviews of negative example suggestions. Note that we ignore the middle rate, which corresponds to a neutral opinion. We ignore negative reviews of positive example suggestions and positive reviews of negative example suggestions since they are not supposed to contain any useful information as they do not share the same perspective about a particular place.

As classifiers we used Support Vector Machine (SVM) and Naïve Bayes classifier. We consider the TF-IDF score for each term as our feature vector, since it indicates the importance of each term to the users. Moreover, it provides a good means to filter out off-topic and noisy terms from reviews. In short, given a user  $u$  and candidate suggestion  $p$ , the similarity score between them,  $S_{BM}(u, p)$ , is the value of the decision function of the SVM classifier or the confidence score of the Naïve Bayes classifier.

### 3.3 Venue Ranking

To rank venues for each user, we combine all scores described above. We calculate a linear combination of all the scores for each  $\langle user, venue \rangle$  pair. The similarity score between a user,  $u$ , and a venue,  $v$ , is calculated as follows:

$$SIM(u, v) = \alpha \times S_{CM}^{Yelp}(u, v) + \beta \times S_{CM}^{TAdvisor}(u, v) + \eta \times S_{TM}(u, v) + \gamma \times S_{BM}(u, v) \quad (2)$$

where  $S_{CM}^{Yelp}(u, v)$  and  $S_{CM}^{TAdvisor}(u, v)$  are the scores based on the categories from Yelp and TripAdvisor, respectively.  $S_{TM}(u, v)$  is the score achieved with Foursquare’s taste keywords, and  $S_{BM}(u, v)$  is the score computed using reviews of users (see Section 3.2). The weights  $\alpha$ ,  $\beta$ ,  $\eta$ , and  $\gamma$  are assigned to the scores to balance the impact of each of them in the final similarity. Finally, for each user  $u$  the venues are ranked based on  $SIM(u, v)$  similarity score.

## 4 Experiments

This section describes the dataset, the experimental setup for assessing the performance of our methodology, and the experimental results.

### 4.1 Dataset and Experimental Setup

Our experiments were conducted on the collection provided by the Text REtrieval Conference (TREC) for the Batch Experiments of the 2015 Contextual Suggestion Track<sup>2</sup>. This track was originally introduced by the National Institute of Standards and Technology (NIST) in 2012 to provide a common evaluation framework for participants that are interested in dealing with the challenging problem of contextual suggestions and venue recommendation.

In short, given a set of example places as user’s preferences (profile) and contextual information (e.g., the *city* where the venues should be recommended), the task consists in returning a ranked list of 30 candidate places which match the user’s profile. Regarding the user context, it may contain the following information: trip type (business, holiday, or other), trip duration (night out, day trip, weekend trip, or longer), group type (alone, friends, family, or other), and season (winter, summer, autumn, or spring). Moreover, user’s age and gender may also be included. While the user profiles consist of a list of venues a particular user has already rated. The ratings range between 0 (very uninterested) and 4 (very interested).

The collection, provided by TREC, consists of a total 9K distinct venues and 211 users. For each user, the contextual information plus a history of 60 previously rated attractions are provided. Additionally, for our experiments, we gathered information about the venues and their corresponding reviews from

<sup>2</sup> <https://sites.google.com/site/trecontext/trec-2015>

**Table 2.** Results for our methods compared with other competitors and TREC median scores. *CatRev-SVM* denotes our submitted system which uses SVM classifier and *CatRev-NB* denotes our submitted system which uses Naïve Bayes classifier

Approach	P@5 Rank	P@5	MRR
<b>CatRev-SVM</b>	<b>1</b>	<b>0.5858</b>	<b>0.7404</b>
CatRev-NB	7	0.5450	0.6991
BASE1	2	0.5706	0.7190
BASE2	3	0.5583	0.6815
TREC Median		0.5090	0.6716

three LBSNs. In particular, we extracted the venues’ categories from Yelp and TripAdvisor, the taste keywords from Foursquare, and the reviews from Yelp.

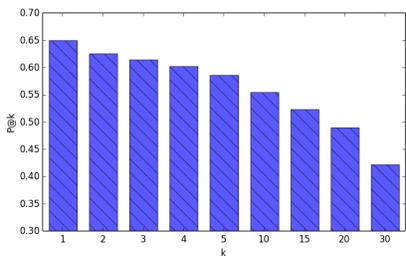
Given a user and a list of 30 candidate suggestions, the recommendation system ranks them. Such generated ranking is then evaluated using relevance assessments, which provide information about whether a given candidate suggestion is relevant to a user or not.

Our ranking of recommendations is done as described in Section 3.3. In order to find the optimum setting for the weights associated with each score of Equation 2, we conducted a 5-fold cross validation that leads to the following setting:  $\alpha = 1.0$ ,  $\beta = 0.3$ ,  $\eta = 0.3$ , and  $\gamma = 1.0$ . As we can see from the values of the weights, Yelp dataset is more significant than TripAdvisor for the categories, and the opinion-based model has a bigger impact on the score, as well.

## 4.2 Results and Discussions

We demonstrate the effectiveness of our model by reporting and analyzing in details the official results of the TREC 2015 Contextual Suggestion Track [6]. We report the performance of our models as well as the two top ranked models reported in the track, briefly comparing the approaches.

The first model is an approach based on collaborative filtering (BASE1) presented in [11]. More specifically, they use factorization machine for venue recommendation. The instances which are fed into the factorization machine are composed of three blocks representing user, context, and venue features. The second one is a similarity-based approach (BASE2) presented in [14]. They create profiles for users and venues using reviews and measure the similarity between the profile pairs to rank the venues. We also compare our results with the median performance of all submitted runs to TREC (TREC Median). In Table 2 we report the values of P@5 (precision-at-5) and MRR (Mean Reciprocal Rank) for our two classifiers: Support Vector Machine (CatRev-SVM) and Naïve Bayes (CatRev-NB), and for the competitors. We run t-test for CatRev-SVM and CatRev-NB and the results were statistically significant at  $p < 0.001$ . Note that we could not carry out the t-test for the BASE1 and BASE2 approaches, since we do not have the rankings from the other competitors.



**Fig. 1.** CatRev-SVM: Precision for different values of  $k = 1, 2, 3, \dots, 30$

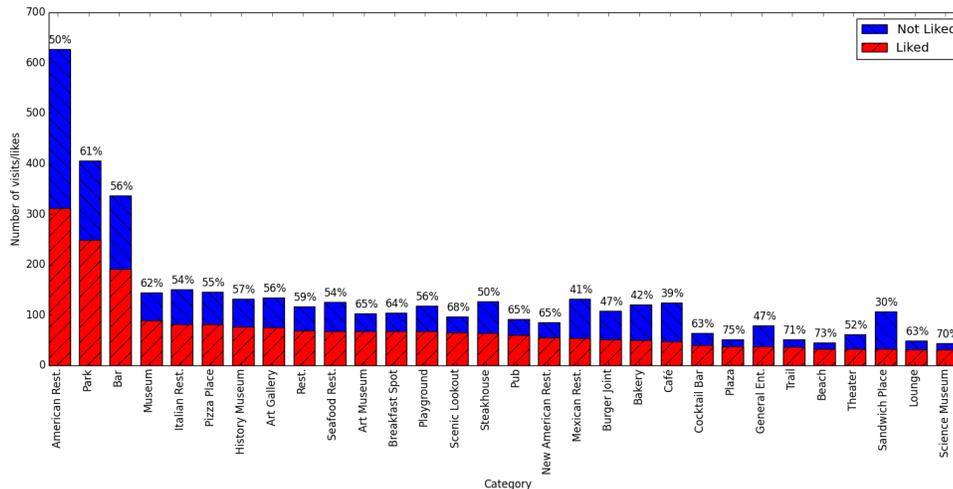
Results in Table 2 demonstrate that both our models perform well compared to TREC median. Specifically, the methodology which utilizes SVM classifier to model a user based on reviews performs best compared to all other submitted runs to TREC and is ranked as top 1 [1,6]. It confirms that our approach of modeling user with reviews from similar users using a machine learning classification algorithm and combining it with other content-based scores is effective for venue recommendation. Better results, however, can be achieved by SVM classifier, since it is more suitable for text classification, which is a linear problem and feature vectors are high dimensional with weights. Moreover, the advantage of linear SVM is that the execution time is very low and there are very few parameters to tune.

It is also worth noting that in several cases there is a lack of negative reviews about venues and the sizes of the positive and negative sets differ significantly. Most of the classification algorithms do not perform well with unbalanced sets, because they tend to correctly classify the class with the larger number of training samples and lower down the overall error rate. However, SVM does not suffer from this, since it does not try to directly minimize the error rate but instead tries to separate the two classes using a hyperplane maximizing the margin. This makes SVM relatively intolerant of the relative size of each class.

In Figure 1 we show the behavior of precision for CatRev-SVM at different  $k = 1, 2, 3, \dots, 30$ . As we can see, the higher precision is achieved with lower  $k$  values, and this is desirable since users on their mobiles are more likely to select a venue on the top of the list.

We report in Figure 2 the distribution of venues over 30 of the most liked types of venues in the dataset. As we can see, the most visited places are *American Restaurant* (10% of the dataset), *Park* (6% of the dataset), followed by *Bar* (5% of the dataset). The figure also shows the number of suggested venues that are liked by the user (the lighter bar). Note that the bars are ordered by their number of likes from left to right.

Following our previous work [12], we calculate a *liked rate* for each type of venue. It is the percentage of suggested venues that are liked by all the users. This percentage is shown on the top of each bar. We could observe that the *Plaza* category is the one with the highest *liked rate* (75%), followed by *Beach* (73%) and *Trail* (71%). Frequently visited categories, such as *American Restaurant* and



**Fig. 2.** Distribution of the number of suggestions the users Liked and Not Liked for the 30 different types of venues. The categories are ordered by the number of liked venues belonging to that particular category. The percentage of liked venues (*liked rate*) of all suggested venues for each category is written on top of their corresponding bar.

*Park*, have a *liked rate* equal to 50% and 61%, respectively. The least categories in term of *liked rate* are *Sandwich Place* (30%) and *Café* (39%). It is also worth noting that according to this figure, the number of users who liked *American Restaurant* is more than *Park*; however, *Park* category has a significantly higher *liked rate* than *American Restaurant*. Note that we cut the long-tail categories, namely, the categories that are not frequently liked, and we did this study only on top 30 liked categories. This study suggests that using a prior probability over categories could potentially benefit a recommender system, and we plan to further explore this direction in a future work.

## 5 Conclusions and Future Work

In this paper we proposed a simple but novel approach for recommending venues. We used frequency-based scores in order to model users' interest and venues, and we enriched the model using users' opinions extracted from reviews written by similar users. Experimental results corroborated the effectiveness of our approach and, although simple, our system managed to outperform all other submitted systems in the TREC 2015 Contextual Suggestion track. This proves the effectiveness of our model compared to state-of-the-art systems under exactly the same settings.

As future work, we would like to propose new scores for other contextual signals that are available in the dataset, such as the *trip type* and *duration*, *group type* and *season*. Furthermore, we would like to enrich the model by including

the preference tags that a user indicates when she rates a venue. One possible way to include them is to find a mapping between them and Foursquare’s taste keywords using an iterative algorithm. Finally, it would be interesting to try different Learning to Rank approaches for combining different scores.

## References

1. M. Aliannejadi, S. A. Bahrainian, A. Giachanou, and F. Crestani. University of lugano at TREC 2015: Contextual suggestion and temporal summarization tracks. In *TREC 2015, Gaithersburg, Maryland, USA*, November 2015.
2. G. Chen and L. Chen. Recommendation based on contextual opinions. In *UMAP, Aalborg, Denmark*, pages 61–73, July 2014.
3. L. Chen, G. Chen, and F. Wang. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, 25(2):99–154, 2015.
4. C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI, Toronto, Canada.*, July 2012.
5. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
6. A. Dean-Hall, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. M. Voorhees. Overview of the TREC 2015 Contextual Suggestion Track. In *TREC, Gaithersburg, USA*, November 2015.
7. S. G. Esparza, M. P. O’Mahony, and B. Smyth. On the real-time web as a source of recommendation knowledge. In *RecSys, Barcelona, Spain*, pages 305–308, September 2010.
8. H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys, Hong Kong, China*, pages 93–100, October 2013.
9. J. Griesner, T. Abdessalem, and H. Naacke. POI recommendation: Towards fused matrix factorization with geographical and temporal influences. In *RecSys, Vienna, Austria*, pages 301–304, September 2015.
10. N. Hariri, Y. Zheng, B. Mobasher, and R. Burke. Context-aware recommendation based on review mining. In *Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, 2011.
11. R. McCreadie, S. Vargas, C. MacDonald, I. Ounis, S. Mackie, J. Manotumruksa, and G. McDonald. University of glasgow at TREC 2015: Experiments with terrier in contextual suggestion, temporal summarisation and dynamic domain tracks. In *TREC, Gaithersburg, USA*, November 2015.
12. A. Rikitienskii, M. Harvey, and F. Crestani. A personalised recommendation system for context-aware suggestions. In *ECIR, Amsterdam, The Netherlands*, pages 63–74, April 2014.
13. X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:421425:1–421425:19, 2009.
14. P. Yang and H. Fang. University of delaware at TREC 2015: Combining opinion profile modeling with complex context filtering for contextual suggestion. In *TREC, Gaithersburg, USA*, November 2015.
15. Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann. Time-aware point-of-interest recommendation. In *SIGIR, Dublin, Ireland*, pages 363–372, July 2013.