Venue Appropriateness Prediction for Personalized Context-Aware Venue Suggestion

Mohammad Aliannejadi and Fabio Crestani
Faculty of Informatics, Università della Svizzera italiana (USI)
Lugano, Switzerland
{mohammad.alian.nejadi,fabio.crestani}@usi.ch

ABSTRACT

Personalized context-aware venue suggestion plays a critical role in satisfying the users' needs on location-based social networks (LBSNs). In this paper, we present a set of novel scores to measure the similarity between a user and a candidate venue in a new city. The scores are based on user's history of preferences in other cities as well as user's context. We address the data sparsity problem in venue recommendation with the aid of a proposed approach to predict contextually appropriate places. Furthermore, we show how to incorporate different scores to improve the performance of recommendation. The experimental results of our participation in the TREC 2016 Contextual Suggestion track show that our approach beats state-of-the-art strategies.

1 INTRODUCTION

With the availability of location-based social networks (LBSNs), such as Yelp, TripAdvisor, and Foursquare, users can share check-in data using their mobile devices. LBSNs collect valuable information about users' mobility records with check-in data including user context and feedback, such as ratings and reviews. Being able to suggest personalized venues to a user, taking into account the user's context, plays a key role in satisfying the user needs on LBSNs, for example when exploring a new venue or visiting a city [6].

There are a number of different LBSNs that are widely used. However, a single LBSN does not have a comprehensive coverage over all venues and all types of information. Moreover, combining user's current context with multimodal information, e.g., ratings and reviews of previously visited venues from different LBSNs, improves the accuracy of venue suggestion [4].

A major challenge for venue suggestion is how to model the user profile, that should be built based on the user feedback on previously visited places. Relevant studies propose to model user profiles based on the venues content [9]. Other studies leverage the opinions of users about a place based on online reviews [6].

Another challenge in venue suggestion is how to leverage the contextual information about users to improve suggestion. To this end, the main focus of the TREC Contextual Suggestion track in 2015 and 2016 [11] was to improve the venue suggestion with the aid of contextual suggestion. However, not many successful participants took into account context. This paper describes our attempt to utilize contextual information to enhance the performance of venue suggestion.

In the effort to face these challenges, our main contribution in this paper can be summarized as follows: (**CO1**) We propose a novel method to predict contextually appropriate venues given the user's current context. (**CO2**) We create a dataset to train our model for contextual appropriateness prediction[1]. (**CO3**) We present a set of scores to measure the similarity between a candidate venue and a user profile based on venues contents and reviews. (**CO4**) We investigate two different ways of combining all the proposed context-, content-based similarity scores.

The official results of the TREC 2016 Contextual Suggestion track, as well as the experiment we have done on the dataset, show that our proposed approach outperforms state-of-the-art strategies.

2 RELATED WORK

Much work has been done to show that user data from LBSNs can significantly improve the effectiveness of a context-aware recommender system. Several rating-based collaborative filtering approaches have been proposed in the literature, which are based on finding common features among users' preferences and recommending venues to people with similar interests. These models are usually based on matrix factorization, exploiting check-in data for recommending places, such as the studies reported in [7, 10]. Factorization Machines generalize matrix factorization techniques to leverage not only user feedback but also other types of information, such as contextual information in LBSNs [13]. Also, some studies follow a review-based strategy, building enhanced user profiles based on their reviews [1]. 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. For example, Chen et al. [6] argued that reviews are helpful to deal with the sparsity problem in LBSNs. Our work consists of different similarity scores each of which is aimed at capturing a different aspect of information. More specifically, our work combines information from venues content and online reviews.

Another line of research tries to incorporate the contextual data to enhance the performance of a recommender system. Levi et al. [14] developed a weighted context-aware recommendation algorithm to address the cold start problem for hotel recommendation. More in details, they defined context groups based on hotel reviews and followed a user's preferences in trip intent and hotel aspects as well as the user's similarity with other users (e.g., nationality)

SIGIR '17, August 7-11, 2017, Shinjuku, Tokyo, Japan

DOI: http://dx.doi.org/10.1145/3077136.3080754

[1]The contextual appropriateness dataset, as well as the additional, crawled data are available at http://inf.usi.ch/phd/aliannejadi/data.html

to recommend hotels. Other works focused more on time as context [9, 18]. Yuan et al. [18] proposed a time-aware collaborative filtering approach. More specifically, their proposed approach recommended venues to users at a specific time of the day by mining historical check-ins of users in LBSNs. Deveaud et al. [9] modeled venues popularity in the immediate future utilizing time series. They leveraged the model to make time-aware venue recommendation to users. The limitation in such line of works is that they only consider time and location as the context, whereas in our work we also take into account other aspects of context. Our work distinguishes itself from these approaches by considering not only time and location but also other contextual dimensions.

The TREC Contextual Suggestion track [11] aimed to encourage the research on context-aware venue recommendation. In fact, the task was to produce a ranked list of venue suggestions for each user in a new city, given the user's context and history of preferences in 1-2 other cities. The contextual dimensions were the duration of the trip, the season in which the trip takes place, the type of the trip (i.e., business, holiday, and other), and the type of group with whom the user is traveling (i.e., alone, friends, etc.). These contextual dimensions were introduced in the track in 2015. Since then, among the top runs, few approaches were trying to leverage such information. Yang and Fang [17] introduced some handcrafted rules for filtering venues based on their appropriateness to a user's current context. As they showed in their paper, applying such filters degrades the performance of the system. Hence, we conclude that contextual appropriateness is not a simple case of using some deterministic rules to filter venues.

Manotumruksa et al. [15] proposed a set of categorical, temporal, and term-based features to train a set of classifiers to predict contextual appropriateness of venues. We believe that such features are very specific to their dataset and problem and not all of them can be generalized to similar problems. Moreover, similar to our work, they collected the classification dataset using crowdsourcing, however, since they asked the workers to assess the appropriateness of a specific venue to a particular user context, this could result in biased assessments. In contrast, we make sure that the assessments are not biased and our crowdsourced collection is general enough to be used in other similar problems.

3 CONTEXTUAL APPROPRIATENESS PREDICTION

In this section, we first define the problem of predicting the contextual appropriateness of venues. We then present the set of features on which we train a classifier to predict the appropriate places. Finally, we describe the collection we used to train the classifier.

3.1 Problem Definition

Given a set of venues $V = \{v_1, \ldots, v_n\}$ and contextual information $C_x = \{cx_1, \ldots, cx_m\}$, the task is to predict if a venue $v_i \in V$ is appropriate to be visited by a user with context C_x. Contextual information expresses user's requirements or preferences and is limited to 3 of those introduced in TREC 2015 Contextual Suggestion track: *Trip type* (business, holiday, other), *Trip duration* (night out, day trip, weekend trip, longer) and *Group type* (alone, friends, family, other). Group type expresses the type of group user prefers to go out with, while trip duration indicates how long the trip will last. We formulate the problem as a binary classification problem. Given a venue v_i and the set of its corresponding categories $C_{v_i} = \{c_1, \ldots, c_k\}$, we consider each category $c_j \in C_{v_i}$ and the aforementioned contextual dimensions as features for classification. The classifier predicts whether the venue category is appropriate to be visited or not.

3.2 Contextual Features

In this section, we discuss the contextual features we used to train a classifier to predict contextual appropriateness of venues. As features, we consider the appropriateness of each contextual dimension to a venue category. For example, assume we have a venue with category *nightlife-spot*, and we want to see if it is appropriate for an example context: trip type: holiday, group type: family, trip duration: weekend. We take the degree of appropriateness of the venue category with each of the contextual dimensions as features. Let $F_{\text{app}}(cat, cxt)$ be a function returning the degree of appropriateness of a venue category, cat, to a contextual dimension, cxt, ranging from -1 to $+1$. $F_{\text{app}}(cat, cxt) = -1$ indicates that a venue with category cat is absolutely inappropriate to be visited by a user with context cxt, whereas $F_{\text{app}}(cat, cxt) = +1$ indicates it is absolutely appropriate. Therefore, the features in this example would be: F_{app}(nightlife-spot, holiday-trip), F_{app}(nightlife-spot, with-family), and F_{app}(nightlife-spot, weekend-trip).

Determining such features may seem intuitive, however, we argue that in many cases determining the output of function $F_{\text{app}}(cat, cxt)$ can be very challenging. For instance, in the previous example, the features can be defined intuitively. On the other hand, defining such features is not as intuitive as other ones: F_{app}(office, with -friends), F_{app}(food-and-drink-shop, business-trip), or F_{app}(stadium, night-out-trip). Based on this observation, we classify the features into two classes: objective and subjective. As the terms suggest, *objective* features are those that are easily determined and are more objective. Therefore, they can influence a user's decision of visiting a venue or not. As in the previous example, supposedly, going to a nightlife spot with a family is *not* appropriate; hence, the user would not go to a nightlife spot even though he/she might like nightlife spots in general. Therefore, *objective* features have a direct impact on users' decisions. *Subjective* features, in contrast, are less discriminative for they depend on each user's opinion and personal preferences.

In the effort to classify the features into *objective* and *subjective* and determine the degree of their objectivity/subjectivity, we designed a crowdsourcing task. In the task, we asked the workers to assess the features. More in detail, we showed them a venue category and a context dimension (e.g., cat = nightlife spot and cxt = Group type: Family) and asked them to assess whether the pair is appropriate or not. We assigned at least five assessors for each category-context pair. The outcome of the task was very interesting since we observed that the workers agreed on one answer when the task was objective, whereas they could not agree on subjective tasks. In this context, those pairs with high agreement rate between the assessors could be considered as *objective*, while those lacking assessors agreement could be seen as *subjective*. Table 1 lists some subjective and objective features in our dataset.

Table 1: Example contextual features

Category	Context	$F_{\mathbf{app}}(cat, cxt)$
Beach	Trip type: Holiday	+1
Zoo	Trip type: Business	−1
Museum	Trip type: Business	−0.66
Pet Store	Trip duration: Weekend	−0.18
Medical Center	Trip type: Other	0.0

As we can see in Table 1, the lower rows are more subjective pairs, in fact, we assign the 0.0 score to the last row. As we discussed earlier, we cannot determine if a venue is appropriate based on such subjective features and therefore we treat them as neutral. We computed the contextual features for all pairs of 11 contextual dimensions and the 177 most frequent categories of Foursquare category tree[2]. Overall we generated 1947 contextual features, leading to 11487 judgments. In fact, this dataset is general enough to be applied to other venue recommendation collections since it considers only place categories, including the most frequent category-context pairs. More details can be found in [3].

3.3 Training the Classifier

As described in Section 3.1, we formulate the problem of contextual appropriateness of venues as a binary classification. We described the features that we use in the classifier in Section 3.2. In this section, we describe how we created the training dataset to train the classifier using the features we created. As training set, we randomly picked 10% of the data from TREC 2016 dataset. To annotate the data, we created another crowdsourcing task in which we asked workers to assess if a category (e.g. Bar) is appropriate to be visited by a user with a specific context (e.g. Holiday, Friends, Weekend). We assigned at least three workers to assess each row in the dataset. Each row was considered appropriate only if at least two of the three assessors agreed on it [3]. Therefore, we trained the contextual appropriateness classifier on 10% of the data from TREC 2016 to predict the rest. As classifier, we trained some widely used classifier, but since we got the best results using Support Vector Machines (SVM) [8], we only report the results of this classifier in this work due to space limitations.

4 CONTEXT-AWARE VENUE SUGGESTION

In this section, we describe our approach of combining context-based and user-based similarity scores to produce a ranked list of venues which fits users' preference and context.

Context-Base Score. As context-based score (denoted as S^F_{cxt}), we consider the value of SVM decision function described in Section 3. If a venue has more than one category, we will run the classification against each category. Then, we consider the minimum value of the decision functions as the context-based similarity score because we observed whenever there are some venue categories from which one is not appropriate to the user's context, it acts as a barrier and leads to the inappropriateness of the venue.

[2] https://developer.foursquare.com/categorytree

Frequency-Based Scores. The other set of scores is based on the frequencies of venue categories and taste tags. We first explain how to calculate the score for venue categories. The score for tags is calculated analogously. Given a user u and a her history of rated venues, $h_u = \{v_1, \ldots, v_n\}$, to each venue is assigned a list of categories $C(v_i) = \{c_1, \ldots, c_k\}$. We define the category index of a user as follows:

Definition 4.1. A *Category Index* consists of categories of venues a user has visited and their normalized frequency in a particular user's profile. The *c*ategory *f*requency (cf) is divided into two sub-frequencies: *positive* category frequency, denoted as cf^+, and *negative* category frequency, denoted as cf^-, representing the normalized frequency of a specific category positively rated and negatively rate by the user, respectively.

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

$$S_{cat}(u, v) = \sum_{c_i \in C(v)} \mathrm{cf}^+_u(c_i) - \mathrm{cf}^-_u(c_i) \ . \qquad (1)$$

We calculated the category similarity score from two sources of information, namely, Foursquare (S^F_{cat}) and Yelp (S^Y_{cat}).

Venue Tags Score. As another frequency-based score, we index the venue taste tags from Foursquare following Definition 4.1. Venue taste tags are the most salient words extracted from the users' reviews. We leveraged them to have a crisper description of the places and improve our suggestions. The tag similarity score is then calculated similarly to Equation 1.

Review-Based Score. A further score uses the reviews to understand the motivation of the user behind a positive or negative rate. Indeed, modeling a user solely on venue's content is very general and does not allow to understand the reasons why the user enjoyed or disliked a venue. Our intuition is that a user's opinion regarding an attraction could be learned based on the opinions of other users who gave the same or similar rating to the same attraction [1]. We created two TF-IDF indexes of reviews per user of venues a user has visited per user: 1) *positive review index* containing only positively rated reviews of venues that a particular user likes, 2) *negative review index* containing only negatively rated reviews of places that a particular user does not like. For each user, we trained a binary classifier considering the positive and negative review indexes as positive and negative training examples, respectively[3]. As classifier, we used SVM with linear kernel and consider the value of the SVM's decision function as the score since it gives us an idea on how close and relevant a venue is to a user profile[4]. We used the reviews from Yelp for this score and refer to it as S^Y_{rev}.

Venue Ranking. We investigated two possible methods of combining these similarity scores: linear interpolation and learning to rank. Linear interpolation is an effective yet simple way of combining multiple scores into one. We linearly combined all the similarity scores into one [2]. We also adopted learning to rank techniques as they have proved to be effective in similar problems [4]. We

[3] An alternative to binary classification would be a regression model, but we believe it is inappropriate since when users read online reviews, they make their minds by taking a binary decision (like/dislike).

[4] Note that we used other well known classifier but do not report the results due to space limitation.

Table 2: Performance evaluation on TREC 2016 Contextual Suggestion track. Bold values denote the best scores.

	P@5	nDCG@5	MAP	MRR	P@10
BASE1	0.4724	0.3306	0.4497	0.6801	0.4552
BASE2	0.5069	0.3281	0.4536	0.6501	0.4500
CS-Linear	0.5069	0.3265	0.4590	0.6796	0.4603
CS-L2Rank	**0.5379**	**0.3603**	**0.4682**	**0.7054**	**0.4828**

adopted ListNet [5] to rank the venues using non-contextual scores as features of ListNet. The contextual score was then used to rerank the initial ranked list to produce the final list of venues. To find the optimum setting for the parameters of our models, we conducted a 5-fold cross validation, training the model using four folds and tuning the parameters using one fold. We denote our linear ranking model as *CS-Linear* and the one based on learning to rank as *CS-L2Rank*.

5 EXPERIMENTAL RESULTS

In this section we report the experimental results to demonstrate the effectiveness of our approach.

Dataset. We report the result of our participation [2] in the TREC 2016 Contextual Suggestion track [11] as well as additional experiments carried out on our crawled dataset[5] [3] and ground truth labels as released by the coordinators.

Evaluation protocol. We follow the same evaluation metrics as in TREC 2016 to report the results, namely, P@5, nDCG@5, MAP, MRR, and P@10.

Compared methods. We compare our approach with top performing systems in TREC 2016. In particular, *BASE1* adopts a modified Rocchio classifier to rank the venues given a query which is created by Rocchio relevance feedback method from places' descriptions and metadata [12]. *BASE2*, on the other hand, considers both the global trend and personal preference to recommend venues. The former is a regressor trained using the most visited category in the 2015 TREC dataset, while the latter adopts word embedding to capture individual user preferences [16].

Results. Table 2 demonstrates the performance of our models against competitors for the TREC 2016. Table 2 shows that CS-L2Rank outperforms the competitors w.r.t. the five evaluation metrics. This indicates that the proposed approach for joint personal-contextual venue suggestion improves the performance of venue suggestion. This happens because our model predicts the contextual appropriateness of venues effectively. At the same time, it improves the ranking technique by capturing user preferences more accurately, thus addressing the data sparsity problem for venue suggestion. CS-Linear, however, beats the baselines w.r.t. MAP and P@10. It exhibits a comparable performance in terms of other evaluation metrics. It also confirms that the proposed similarity scores are able to capture contextual appropriateness and user interest. However, it indicates that combining multimodal information is a complex problem and thus more sophisticated techniques, such as learning to rank, perform better.

[5] Available at http://inf.usi.ch/phd/aliannejadi/data.html

6 CONCLUSION

In this study, we presented an approach to predicting contextually appropriate venues as well as other similarity scores to model the personal preferences of users for venue suggestion. For contextual appropriateness prediction, we proposed a set of novel relevance features with which we trained a classifier. The features as well as the training data was created using crowdsourcing and is freely available on request. We studied two directions to combine the scores: linear interpolation and learning to rank. The proposed CS-L2Rank model exhibited the best performance beating state-of-the-art approaches in terms of all five evaluation metrics. This confirms that the proposed approach, CS-L2Rank, solves the data sparsity problem and captures user context and preferences more accurately. As future work, we plan to extend our model to capture the time dimension and perform time-aware venue suggestion.

ACKNOWLEDGMENTS

This research was partially funded by the RelMobIR project of the Swiss National Science Foundation (SNSF).

REFERENCES

[1] Mohammad Aliannejadi, Ida Mele, and Fabio Crestani. 2016. User Model Enrichment for Venue Recommendation. In *AIRS*. Springer, 212–223.
[2] Mohammad Aliannejadi, Ida Mele, and Fabio Crestani. 2016. Venue Appropriateness Prediction for Contextual Suggestion. In *TREC*. NIST.
[3] Mohammad Aliannejadi, Ida Mele, and Fabio Crestani. 2017. A Cross-Platform Collection for Contextual Suggestion. In *SIGIR 2017*. ACM.
[4] Mohammad Aliannejadi, Dimitrios Rafailidis, and Fabio Crestani. 2017. Personalized Keyword Boosting for Venue Suggestion Based on Multiple LBSNs. In *ECIR*. Springer, 291–303.
[5] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *ICML*. ACM, 129–136.
[6] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction* 25, 2 (2015), 99–154.
[7] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. 2012. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks. In *AAAI*. AAAI Press, 17–23.
[8] Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine Learning* 20, 3 (1995), 273–297.
[9] Romain Deveaud, M-Dyaa Albakour, Craig Macdonald, and Iadh Ounis. 2015. Experiments with a Venue-Centric Model for Personalised and Time-Aware Venue Suggestion. In *CIKM*. ACM, 53–62.
[10] Jean-Benoît Griesner, Talel Abdessalem, and Hubert Naacke. 2015. POI Recommendation: Towards Fused Matrix Factorization with Geographical and Temporal Influences. In *RecSys*. ACM, 301–304.
[11] Seyyed Hadi Hashemi, Charles L. A. Clarke, Jaap Kamps, Julia Kiseleva, and Ellen M. Voorhees. 2016. Overview of the TREC 2016 Contextual Suggestion Track. In *TREC*. NIST.
[12] Georgios Kalamatianos and Avi Arampatzis. 2016. Recommending Points-of-Interest via Weighted kNN, Rated Rocchio, and Borda Count Fusion. In *TREC*. NIST.
[13] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.
[14] Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. 2012. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In *RecSys*. ACM, 115–122.
[15] Jarana Manotumruksa, Craig MacDonald, and Iadh Ounis. 2016. Predicting Contextually Appropriate Venues in Location-Based Social Networks. In *CLEF*. Springer, 96–109.
[16] Jian Mo, Luc Lamontagne, and Richard Khoury. 2016. Word embeddings and Global Preference for Contextual Suggestion. In *TREC*. NIST.
[17] Peilin Yang and Hui Fang. 2015. University of Delaware at TREC 2015: Combining Opinion Profile Modeling with Complex Context Filtering for Contextual Suggestion. In *TREC*. NIST.
[18] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat-Thalmann. 2013. Time-aware point-of-interest recommendation. In *SIGIR*. ACM, 363–372.