On the Importance of Venue-Dependent Features for Learning to Rank Contextual Suggestions

Romain Deveaud M-Dyaa Albakour Craig Macdonald Iadh Ounis
University of Glasgow, UK
firstame.lastname@glasgow.ac.uk

ABSTRACT

Suggesting venues to a user in a given geographic context is an emerging task that is currently attracting a lot of attention. Existing studies in the literature consist of approaches that rank candidate venues based on different features of the venues and the user, which either focus on modelling the preferences of the user or the quality of the venue. However, while providing insightful results and conclusions, none of these studies have explored the relative effectiveness of these different features. In this paper, we explore a variety of user-dependent and venue-dependent features and apply state-of-the-art learning to rank approaches to the problem of contextual suggestion in order to find what makes a venue relevant for a given context. Using the test collection of the TREC 2013 Contextual Suggestion track, we perform a number of experiments to evaluate our approach. Our results suggest that a learning to rank technique can significantly outperform a Language Modelling baseline that models the positive and negative preferences of the user. Moreover, despite the fact that the contextual suggestion task is a personalisation task (i.e. providing the user with personalised suggestions of venues), we surprisingly find that user-dependent features are less effective than venue-dependent features for estimating the relevance of a suggestion.

Categories and Subject Descriptors: H.3.3 [Information Storage & Retrieval]: Information Search & Retrieval

Keywords: Venue recommendation; contextual suggestion; learning to rank; personalisation

1. INTRODUCTION

The ever increasing popularity of mobile devices, coupled with ubiquitous Internet access, allows people to search for information in almost every situation and at every hour of the day. As a consequence, search is becoming increasingly local, where people issue queries that are related to their surroundings, mainly for entertainment purposes [8] (e.g. finding a restaurant or activities for the afternoon). Directly suggesting informational content to the users without requiring them to issue a query (i.e. zero-query retrieval) has recently been identified as one of the major Information Retrieval (IR) research directions, according to the report of the SWIRL 2012 workshop [2].

The TREC Contextual Suggestion track [5] explores such a task and provides a common evaluation framework, allowing researchers to propose solutions aimed at tackling the wide range of challenges associated recommending venues in a city [6]. The aim of the task is to return a ranked list of suggestions (venues) that are relevant given the geographical context (a location in a city) of the users and their preferences. Successful TREC participants [7, 13, 14, 18] relied on the public API of travel sites (such as Foursquare, Yelp, or Google Places) to identify popular and interesting venues, and to filter out suggestions that do not satisfy these geographical constraints. Hence, one of the key challenges of the TREC track is to model the interests of the users, by making use of the preferences they indicated in their profile, and thereby provide them with a ranked list of personalised suggestions. This problem has been mainly tackled using content-based recommendation approaches, considering either the categories of the venues [7], the descriptions of example venues provided by the track organisers [13, 14], or the reviews entered by users on various travel sites for these venues [18].

However, while all of these preceding approaches have deployed useful ranking features, none has tried to combine them together into a single ranking model. In this paper, we propose to learn models that can take all these different features into account, and to explore their effectiveness with the aim of discovering what makes a contextual suggestion relevant. To this end, we define 64 different features before applying learning to rank techniques [10], and perform a thorough evaluation using the test collection of the TREC 2013 Contextual Suggestion track. The contributions of this paper are two-fold. Firstly, we experiment with several state-of-the-art learning to rank techniques for contextual suggestion and show that, while the models learned with the complete set of features can outperform a Language Modelling baseline [14] by up to 77% in terms of P@5, user-dependent features are surprisingly not as important as venue-dependent features for estimating the relevance of a venue. Secondly, we conduct an investigation of the importance of each of the venue-dependent features and find that the probability that a venue is "liked" or "tipped"[1], given a city, is the most prominent indicator of relevance.

[1]"Tips" are the equivalent of user reviews in Foursquare.

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2661956 .

2. LEARNING TO RANK CONTEXTUAL SUGGESTIONS

The goal of the following experiments is to learn robust and effective models for ranking contextual suggestions. The problem investigated by the TREC 2013 Contextual Suggestion track [5] is to return a list of ranked suggestions, given a context and the profile of a user. The context can be one of the 50 American cities considered in the 2013 dataset, and the user's profile is built by asking the user to provide their preferences, through 5-point numerical ratings (0 to 4), for 50 example venues in Philadelphia, PA. In the Contextual Suggestion track, suggestions are represented by the Web pages of venues that can be extracted from the open Web. To remove the confounding variable of geographically irrelevant venues, we filter out the suggestions that are not relevant to any of the contexts. Most of the TREC participants followed the same approach when they used the API of travel sites to filter geographically irrelevant venues [7, 14, 18]. Using the Terrier IR platform [12], we then build an index containing the Web pages of these venues. For each *query* (i.e. a pair of user & context) from a total of 223 pairs in the dataset, we produce an initial *sample* of candidate Web pages that we will further re-rank through our learning to rank approach. This sample is composed of 216 venues on average and is generated using a Language Modelling (LM) baseline, which favours venues that are similar to the highly rated venues in the user's profile (rated with 3 or 4) and that are dissimilar to the poorly rated venues (rated with 0 or 1). We specifically use the technique detailed in [1], without the diversification step. This technique was found to be effective in previous work [14], and similar statistics were used in other successful approaches [18].

We adopt a learning to rank approach in this paper, hence we compute several features for each suggestion retrieved in the LM sample. Previous work has showed that the popularity of a venue [9] – represented by its all-time number of visitors, or *check-ins* – is a strong indicator of relevance, but such an attribute cannot be obtained directly from the venues' Web pages. However, travel sites or Location-Based Social Networks (LBSNs) such as Foursquare or Yelp allow to obtain such information about the venues. We automatically map Web pages to Foursquare venues by combining and intersecting the results of the Google and Foursquare search APIs when issuing a query formed by the title of the Web page (which always contains the name of the venue) and the name of its city. This method allows us to retrieve all the information and attributes provided by Foursquare, and to link them to the suggestion Web pages of our index. By performing a manual evaluation on a random subset of 100 suggestions, we observed that 87% of them were correctly mapped to their entry in Foursquare. However, we also noticed that only 57% of the relevant suggestions (according to the relevance judgments) have been associated with a Foursquare venue. Tackling this problem is out of the scope of this paper, but we plan to address it in future work by integrating several other LBSNs and Linked Open Data sources.

We then calculate a set of 64 features using the information obtained from Foursquare for the Web pages of the LM sample. These features can be divided into four different groups: 25 city-dependent (City), 20 category-dependent (Cat), 10 venue-dependent (Venue), and 9 user-dependent (User) features.

City: These features describe the context and they include the number of venues, and the total number of checkins, likes, tips, and photos in the city. We also consider the minimum, maximum, average, median, and standard deviation of these four last attributes across the venues of the city.

Cat: The category-dependent features consist of the counts of the 10 highest level Foursquare categories of the venue[2], as well as the same counts using only the venues that the user labelled as relevant in her/his profile (rated with 3 or 4).

Venue: Venue-dependent features are mostly related to the popularity of the venue, including its number of checkins, likes, tips, and photos entered by Foursquare users. Since explore the importance of each of these features in our experiments, we provide further details and a complete description of the features in Table 1.

Table 1: Description of the venue-dependent Foursquare features (Venue) used in this work.

Feature name	Description
NbCheckins	Total number of check-ins in the venue.
NbLikes	Total number of "likes" for the venue.
NbTips	Total number of "tips" for the venue.
NbPhotos	Total number of photos that have been taken in the venue.
Rating	Average of all the ratings given by the users for the venue.
CheckinRatio	$\frac{\text{NbCheckins}}{\text{NbCheckinsInCity}}$
LikeRatio	$\frac{\text{NbLikes}}{\text{NbLikesInCity}}$
TipRatio	$\frac{\text{NbTips}}{\text{NbTipsInCity}}$
PhotoRatio	$\frac{\text{NbPhotos}}{\text{NbPhotosInCity}}$
Distance	Distance of the venue from the center of the city.

User: The selected user-dependent features reproduce approaches that several studies and TREC participants have proposed for personalising the suggestions. Firstly, we consider the matches between the categories of the venue and the categories of the user [7] by computing the cosine similarity between the two vectors of category counts computed for the Cat features. We also compute another cosine similarity which considers the categories of the venues that the user did not like. Secondly, we consider the text description of the example venues, as well as the "tip" reviews entered by the Foursquare users, to build two textual user profiles [14]: a positive one generated from the example venues that the user rated highly (either 3 or 4), and a negative one (constructed from the example venues rated by either 0 or 1). Both these profiles are represented as term vectors. Using these profiles, we compute the cosine similarity between the term vector of the venue (generated from its tip reviews) and the positive and negative user profiles respectively. Furthermore, we consider the polarity of the tip reviews to generate four more features. Using the SentiStrength [15] sentiment analysis tool, we classify all of the tip reviews of the venues into three different classes: positive, negative, and neutral. Following this, we construct another positive user profile using the positive reviews of the example venues they rated highly. Likewise, the other negative profile is constructed from the negative reviews of the example venues the user rated poorly. As a result, four features of cosine similarity are generated from the combinations of the user profiles

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

Table 2: Contextual suggestion effectiveness results for the different learning to rank models, as well as for the ablated groups of features. All the models learned with the set of 64 features exhibit statistically significant improvements over the initial ranking (LM baseline) according to a paired t-test ($p < 0.01$). Significant decreases induced by features ablations are indicated by ▼, also according to a paired t-test ($p < 0.01$).

	P@5		P@10		MRR	
Initial ranking [1] (LM)	0.2099		0.1910		0.3660	
AFS [11] (All)	**0.3148**		0.2874		**0.5446**	
- City	0.3058	(-2.85%)	0.2848	(-0.94%)	0.5418	(-0.51%)
- Cat	0.3058	(-2.85%)	**0.2888**	(+0.47%)	0.5346	(-1.83%)
- User	0.3031	(-3.70%)	0.2794	(-2.81%)	0.5308	(-2.53%)
- Venue	0.3058	(-2.85%)	0.2744	(-4.52%)	0.5332	(-2.08%)
Adarank [17] (All)	**0.2735**		0.2565		0.4794	
- City	0.2709	(-0.98%)	0.2623	(+2.27%)	**0.4857**	(+1.31%)
- Cat	0.2610	(-4.59%)	**0.2713**	(+5.77%)	0.4717	(-1.61%)
- User	0.2556	(-6.56%)	0.2435	(-5.07%)	0.4450	(-7.18%)
- Venue	0.2458	(-10.13%)	0.2401	(-6.38%)	0.4423	(-7.74%)
RankNet [3] (All)	**0.2816**		0.2610		0.4648	
- City	0.2726	(-3.18%)	**0.2673**	(+2.41%)	**0.4665**	(+0.37%)
- Cat	0.2547▼	(-9.55%)	0.2502	(-4.12%)	0.4623	(-0.52%)
- User	0.2559▼	(-9.15%)	0.2484	(-4.81%)	0.4401	(-5.31%)
- Venue	0.2574▼	(-8.60%)	0.2507	(-3.95%)	0.4487	(-3.45%)
LambdaMART [16] (All)	0.3713		0.3211		0.6093	
- City	0.3668	(-1.21%)	0.3256	(+1.40%)	0.5874	(-3.59%)
- Cat	0.3570	(-3.86%)	0.3233	(+0.70%)	0.5918	(-2.87%)
- User	**0.4009**	(+7.97%)	**0.3386**	(+5.45%)	**0.6584**	(+8.06%)
- Venue	0.2960▼	(-20.29%)	0.2691▼	(-16.20%)	0.5348▼	(-12.22%)

and the venue's positive or negative reviews. The intuition behind these features, which showed good performances on the 2012 Contextual Suggestion dataset [18], is that people with similar opinions about why would they like or dislike a venue would have similar tastes, and vice versa. Our last User feature estimates the variation in the diversity of interests between users and is estimated using the entropy of category probability distribution for a given user, from the top level categories in Foursquare of the venues they like. A low-entropy user is then likely to be interested in a few types of venues (e.g. only museums), while a high-entropy user is likely to be open to a wide range of suggestions.

We re-rank the venues of the LM sample and explore the effectiveness of four different learning to rank techniques: Automatic Feature Selection (AFS) [11], Adarank [17], RankNet [3], and LambdaMART [16][3]. So as to ascertain the effect of each group of features, all of these models are first learned using the aforementioned 64 features, then learned again after ablating one group of features at a time. Our experiments are conducted using a 5-fold cross validation across the 223 pairs of user/context of the TREC 2013 Contextual Suggestion track for which contextual suggestions have been judged. Each fold has separate training, validation, and test sets. We report the results of our learning to rank experiments in the following section.

3. EXPERIMENTAL RESULTS

For each group of feature (City, Cat, User, or Venue), we remove it from the set of 64 features and learn a ranking model. By performing such an ablation, we can explore the importance of each group of features and derive some insights on their impact on the ranking of suggestions. We remove the groups of features independently from each other: no more than one group of features is removed at the same time. We show the effectiveness results of all of the learned models (AFS, Adarank, RankNet, and LambdaMART) and the results obtained for all feature group ablations in Table 2. Rows with (All) correspond to models that have been learned using the full set of 64 features. On analysing this table, we see that RankNet and AFS are similarly degraded by the removal of feature groups. On the other hand, Adarank and LambdaMART in particular (which is actually the best performing model in our experiments) exhibit their largest decreases in performance when removing the venue-dependent features from the features set. This suggests that, for these models, popular venues constitute relevant suggestions, even for a personalised task such as the TREC Contextual Suggestion track.

In particular, we observe that the best overall results are achieved by the LambdaMART technique, which already showed strong performance for Web search by winning the 2011 Yahoo! Learning to Rank Challenge [4]. For LambdaMART, ablating the user-dependent features leads to an ≈8% increase in P@5 (0.3713 → 0.4009) and MRR (0.6093 → 0.6584), and a 5.45% increase in P@10 (0.3211 → 0.3386), which shows that these features can confuse this model. On the other hand, ablating the venue-dependent features causes a statistically significant (t-test, $p < 0.01$) decrease of performance by up to 20.29% in terms of P@5 (0.3713 → 0.2960), showing the great importance of these features for learning an effective ranking model.

While all groups of features seem to play an important role in learning an effective model, venue-dependent features appear to be more important, especially when used with the

[3] `http://code.google.com/p/jforests`

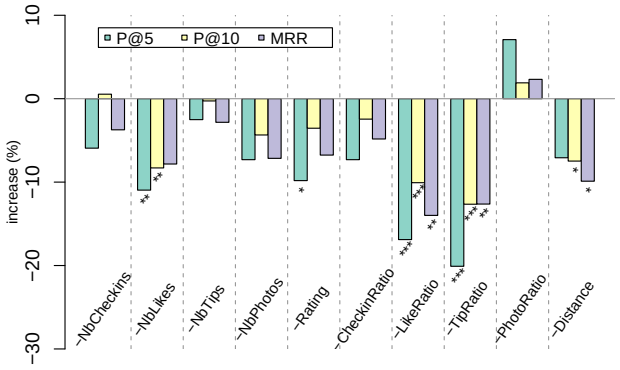

Figure 1: Percentage of improvement obtained when independently removing single venue-dependent features, with respect to a LambdaMART baseline that uses all 64 features. Improvements are expressed in terms of P@5, P@10, and MRR. Statistical significance is stated according to a paired t-test (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$).

best performing learned model. Hence, we conduct another feature ablation experiment to explore the individual effectiveness of these venue-dependent features, in order to determine which single features are the most effective when suggesting venues to users. In this experiment, we consider the LambdaMART ranking model – learned using all the 64 features – as a baseline, and we compare its performances to other LambdaMART models that have been learned after removing each of the venue-dependent features individually.

Similarly to the previous experiment, a decrease in performance implies that the feature is deemed useful. We report the results in Figure 1. The first observation we make is that PhotoRatio appears to be harmful. When Foursquare venues do not have any photo, the value of this feature is equal to zero, which seems to confuse the learner. Likes and tips, which are more abundant and hence do not suffer from this problem, appear to be very strong indicators of relevance. It is important to note that the raw numbers (i.e. NbLikes and NbTips) are not enough, and that using the city context greatly improves the importance of these features (see LikeRatio and TipRatio). The rating of the venue (which is an average of all the ratings provided by Foursquare users) is also a good indicator of relevance, but to a less extent than LikeRatio and TipRatio. Finally, the distance between the venue and the center of the city also seems to play an important role. Since city centres usually are the most vibrant parts, using this distance as a feature allows the learned model to implicitly separate potentially relevant and attractive venues from unpopular ones.

4. CONCLUDING DISCUSSION

While we expected the learned models to take advantage of all groups of features, we observed surprising results, especially concerning LambdaMART (the best performing model) and Adarank, where venue-dependent features were found to be the most important. These results however raise several questions: are users really interested in personalised venue suggestions? If yes, does personalisation depend on other uncontrolled parameters (e.g. tourists vs. residents)? Do these observations result from a bias in the judging process of the Contextual Suggestion track? We can also safely hypothesise that users, who are complex by nature, are more difficult to model than venues that are only described by a handful of attributes from location-based social networks. Nevertheless, the results of this paper open a wide range of research questions that might be interesting to answer in future work.

Acknowledgments

This work has been carried out in the scope of the EC co-funded project SMART (FP7-287583).

5. REFERENCES

[1] M.-D. Albakour, R. Deveaud, C. Macdonald, and I. Ounis. Diversifying Contextual Suggestions from Location-based Social Networks. In *Proc. of IIiX*, 2014.

[2] J. Allan, B. Croft, A. Moffat, and M. Sanderson. 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, May 2012.

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to Rank Using Gradient Descent. In *Proc. of ICML*, 2005.

[4] O. Chapelle and Y. Chang. Yahoo! Learning to Rank Challenge Overview. In *Yahoo! Learning to Rank Challenge at ICML 2010*, 2011.

[5] A. Dean-Hall, C. Clarke, J. Kamps, P. Thomas, N. Simone, and E. Vorhees. Overview of the TREC 2013 contextual suggestion track. In *Proc. of TREC*, 2013.

[6] R. Deveaud, M.-D. Albakour, C. Macdonald, and I. Ounis. Challenges in Recommending Venues within Smart Cities. In *Proc. of i-ASC at ECIR*, 2014.

[7] G. Hubert, G. Cabanac, K. Pinel-Sauvagnat, D. Palacio, and C. Sallaberry. IRIT, GeoComp, and LIUPPA at the TREC 2013 Contextual Suggestion Track. In *Proc. of TREC*, 2013.

[8] H. Kukka, V. Kostakos, T. Ojala, J. Ylipulli, T. Suopajärvi, M. Jurmu, and S. Hosio. This is Not Classified: Everyday Information Seeking and Encountering in Smart Urban Spaces. *Personal Ubiquitous Comput.*, 17(1), 2013.

[9] Y. Li, M. Steiner, L. Wang, Z.-L. Zhang, and J. Bao. Exploring venue popularity in Foursquare. In *Proc. of INFOCOM*, 2013.

[10] T.-Y. Liu. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[11] D. Metzler. Automatic Feature Selection in the Markov Random Field Model for Information Retrieval. In *Proc. of CIKM*, 2007.

[12] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proc. of OSIR at SIGIR*, 2006.

[13] A. Rikitianskiy, M. Harvey, and F. Crestani. University of Lugano at the TREC 2013 Contextual Suggestion Track. In *Proc. of TREC*, 2013.

[14] M. Sappelli, S. Verberne, and W. Kraaij. Recommending Personalized Touristic Sights Using Google Places. In *Proc. of SIGIR*, 2013.

[15] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment Strength Detection for the Social Web. *Journal of the American Society for Information Science and Technology*, 63(1), 2012.

[16] Q. Wu, C. J. C. Burges, K. M. Svore, and J. Gao. Ranking, Boosting, and Model Adaptation. Technical Report MSR-TR-2008-109, Microsoft, 2008.

[17] J. Xu and H. Li. AdaRank: A Boosting Algorithm for Information Retrieval. In *Proc. of SIGIR*, 2007.

[18] P. Yang and H. Fang. Opinion-based User Profile Modeling for Contextual Suggestions. In *Proc. of ICTIR*, 2013.