TNO and RUN at the TREC 2012 Contextual Suggestion Track: Recommending personalized touristic sights using Google Places

Maya Sappelli*[1], Suzan Verberne†[2] and Wessel Kraaij‡[1]

[1]TNO & Radboud University Nijmegen
[2]Radboud University Nijmegen

Abstract

The purpose of the Contextual Suggestion track is to suggest personalized touristic activities to an individual, given a certain location and time. In our approach, we collected initial recommendations by using the location context as search query in Google Places. We first ranked the recommendations based on their textual similarity to the user profiles. In order to improve the ranking of popular sights, we combined the resulted ranking with a number of other rankings based on Google Search, popularity and categories. Finally, we performed filtering based on the temporal context.

1 Introduction

The purpose of the Contextual Suggestion track is to suggest personalized touristic activities to an individual, given a certain spatial-temporal context. For development purposes a set of example activities in Toronto is provided in the form of web sites (title, description and url) that are rated by individuals.

We considered two approaches for this task. The first is using the rated training examples to generate personalized query words in the search for touristic activities. The search results need then be filtered to the context (i.e. removing suggestions that do not match the locational and temporal context). The second approach is to do this the other way around, using the context and a generic term such as "touristic attractions" as a search query and then rerank the search results based on the information extracted from the trainings examples. We chose to work out the latter approach.

*m.sappelli@cs.ru.nl
†s.verberne@cs.ru.nl
‡w.kraaij@cs.ru.nl

2 Data

As input to the task, we were provided with a set of 34 profiles, 49 examples of suggestions and 50 contexts in XML form.

Each profile corresponded to a single user and consisted of a list of rated example suggestions. The rating were divided into an initial rating, based on the title and description of the example and a final rating, based on all available information.

Each example suggestion consisted of a title and a short description of the suggestion as well as an associated URL. The example suggestions were a collection of bars, museums and other touristic activities in the Toronto area.

Each context consisted of spatial information (cityname, statename, latitude, longitude) and temporal information (day, time and season). The day and time were categorical values, weekday or weekend day and morning, afternoon or evening, respectively.

The task was to provide a ranked list of 50 suggestions for each profile/context pair. Each suggestion should contain a title, description and associated URL. The description of the item may be personalized. The suggestions should be appropriate to the profile as well as the geotemporal context. Time-wise, the user has five hours available for the suggestion, which puts limitations on the locations of the suggestions.

3 Method

Out method comprises 5 steps: (1) Collecting the initial search results, (2) building the user profiles, (3) ranking the recommendations for the user profile, (4) re-ranking the list of recommendations, (5) filtering the recommendations using the temporal context.

3.1 Collecting search results

We started off by collecting search results for each context. In our approach we mainly used the Google Places API. The advantage of this API is that the returned search results contain many details such as opening hours, reviews and events. We used the longitude and latitude of the location in the context as input for the Google Places API. We extended this with the keyword “tourist attractions” to focus on touristic activities. The Google Places API was limited in the number of search results it returned. We increased the radius within which to search in each search request, with a maximal radius of 50 km, in order to obtain a sufficient amount of search results for each context (i.e. more than 50 search results). To obtain short descriptions of the search results we made use of the Google Custom Search

API. As input query for the custom search we used the URL of the search result from Google Places or its title when there was no URL available.

3.2 Building the Profiles

We represented the user using a profile consisting of terms with a positive or negative association to the examples the user had rated. We assume that the terms in the examples rated with 1 have a positive association for the user, while the ones rated with -1 have a negative association for the user. For each user we created a “positive” profile and a “negative” profile.

In the provided profiles, there was a distinction between initial and final ratings. The initial rating was based on the title and description of the example only, the final rating included the full content of the web site as well. Sometimes the initial rating was not the same as the final rating. Therefore we used the initial rating to decide in which profile (“positive” or “negative”) the terms from the title and description belonged and the final rating to decide in which profile the terms extracted from the full web site belonged. Terms from title and description with neutral initial ratings, or terms from the web site with neutral final ratings were ignored.

Our initial idea was to simply extract the contents of the web site and use those words in the profile. However, it is not straightforward to extract textual content since most web content is polluted with advertising material or inaccessible because it contains flash for example. Therefore we decided to take a different approach in obtaining textual content for the url. We used the title and url in the example as search terms in the Google Places API. If there was a hit, we extracted the categories, reviews and events for the first hit and used the terms in them for the respective “positive” or “negative” profile. This results in the user model $UM = \{R_p, R_n\}$ in which R_p is the term frequency vector representation of the “positive” profile and R_n of the “negative” term profiles.

3.3 Ranking recommendations

We implemented two variants of our approach: a similarity based method and a scoring based method

3.3.1 Run 1: Similarity based method - run01TI

For each user the terms in UM will be weighted. In the similarity based method we used a tf-idf measure [3] to determine the importance of each term in the profile. The following metric was used to obtain a normalized term frequency for each term:

$$tf(t,p) = \frac{f(t,p)}{max\{f(w,p) : w \in p\}}$$

where $f(t,p)$ is the frequency of term t in profile p and p could either be the set of words from the "positive" profile or from the "negative" profile. The inverse document frequency was obtained using the following formula:

$$idf(t,E) = log\frac{|E|}{|\{e \in E : t \in e\}|}$$

where the total number of examples $|E|$ is divided by the number of examples containing term t. Each term in each profile can now be represented with its tf-idf value using the following formula:

$$tfidf(t,p) = tf(t,p) \cdot idf(t,E)$$

For each search result, the terms in the title, google snippet, reviews and events are extracted. Using the same method as in the aforementioned formulas the tfidf values are calculated for the terms, but the term frequency is of course based on the search result itself, rather than the "positive" or "negative" profile.

Since we now have a vector representation of the search result and vector representations of the "positive" and "negative" profiles, we can calculate the similarity between the search results and the profiles using the cosine similarity measure. The following function is used:

$$cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{||\vec{A}||||\vec{B}||}$$

where $\vec{A}$ is the search result vector and $\vec{B}$ is either the "positive" or the "negative" profile vector. This gives us two similarity values for each search result. One is the similarity to the "positive" profile, the other for the "negative" profile. We can rank the search results based on these similarity scores. We order each items descending on their $cos_{positive}$ score. However, when $cos_{negative} > cos_{positive}$ we place the item at the bottom of the list (i.e. after the item with the lowest $cos_positive$ score, but with $cos_{positive} > cos_{negative}$). We left them in the list of recommendations to be able to meet the number of requested recommendations (i.e 50 recommendations per person/context combination).

3.3.2 Run 2: Language modelling method run02K

We only use the "positive" profile from the user model UM in this method. We take a language modelling approach and use Kullback-Leibler divergence to weigh each term. We use pointwise Kullback-Leibler divergence [2], as suggested by [1]. It functions as a measure of term importance that indicates how important the term is to distinguish the "positive" examples from all examples. The following function was used:

$$KLdiv(t|p) = P(t|p)log\frac{P(t|p)}{P(t|C)}$$

where $P(t|p)$ is the probability of observing term t in the set of “positive” examples and $P(t|C)$ is the probability of observing term t in the set of all examples.

A search result is better when it has many terms that are important in the “positive” examples. For each search result we derive its score by taking the sum of the Kullback-Leibler scores for the terms describing the search result:

$$\sum_{t \in S} KLdiv(t|p)$$

where S is the set of terms describing the search result. The search results are ordered descendingly on their scores.

3.4 Re-ranking the list of recommendations

Since we did not have a proper evaluation set, we evaluated the results of the two methods by rating the example set ourselves and evaluating the recommended items for our own profile intuitively (e.g which order of suggested activity appealed more to us).

We noticed in the suggestions given by the two runs, that famous touristic attractions do not rank very well. This is likely an artifact of the example data. For example, the Statue of Liberty, does not resemble any of the examples in the example attractions in Toronto, so it is no surprise that it does not receive a high rank. However, we believe that these famous sites should rank well. Therefore we use some elements from the Google Places API to increase the rank of these items.

We take an approach in which we created 4 ordered ranked lists. The final rank is determined by the weighted average rank of the search result in these 4 ordered lists. The first list is the ranking obtained from step 3 (similarity or scoring based method). The second list is based on the original Google ranking which is based on the importance of a place, thus ranking more prominent places high. The third list is based on the Google rating that other people have given to the place.

The final list is based on the a priori category likelihood. This is based on the idea that some people have preferences for certain categories of activities (such as museums) rather than preferences for individual items. From the training examples we derived the relevant Google categories such as amusement park, bar, or museum. We then determined the likelihood that a certain category is rated positively by this person. We do this by counting the number of times the category occurs in the set of positive examples and divide it by the total number of category occurrences in the set of positive examples. We applied +1 smoothing to account for categories that do not occur in the example set. For each search result the a priori category likelihood is estimated by taking the average of the a priori category likelihood of the categories for that result.

Each list was weighted equally in the weighted average, except the rank based on the a priori category likelihood, which was weighted with 0.5. This choice was made because many categories did not occur in the example set. Because of this missing information we did not want this ranking to be of too much influence.

3.5 Filtering based on temporal context

In the last phase, we filter the search results that do not match the temporal part of the given context. We use the opening hours as registered in Google Places as reference material for determining whether a result matches the temporal context or not. For example, when the temporal context is evening, we do not suggest search results that have opening hours until 5pm.

3.6 Presentation of the results

The first impression of a search result is very important. However, not all Google snippets gave a good presentation of the suggestion. Sometimes it contained advertisements or an unclear description. Therefore, we decided to use positive reviews as descriptions for the suggested places. Although not all reviews are a good descriptor for the suggestion, we think that the positiveness of the descriptions may prime people to give a good initial rating. If their initial rating is positive, they may be inclined to give a positive final rating as well.

4 Results

In this section we present the accuracy and precision@5 results that we obtained with the two runs we submitted: (1) run01TI ranking based on TFIDF with cosine similarity and (2) run02K ranking based on point-wise Kullback-Leibler divergence scores. There were only 44 out of 1750 profile/context pairs taken into account during evaluation and only the top 5 suggestions were evaluated. All results in this section are based on these 220 (e.g. 44*5) datapoints.

Table 1 shows the precision results for the different measures. Only items that have scored a rating of 2 (i.e good fit, or interesting) are taken into account. It shows that there are only very small differences between the TFIDF measure and the Kullback-leibler divergence measure. The results show a particularly high precision at rank 5 for the geographical fit. The precision for the top 5 ranks in terms of the rating on description and website

Table 1: Precision @5 results for both runs

	Website*GeoTemporal	Description	Website
run01TI	0.1907	0.4185	0.3963
run02K	0.2185	0.4049	0.4710
	Geotemporal	Geo	Temporal
run01TI	0.5392	0.8934	0.5598
run02K	0.5649	0.9034	0.5839

can be improved. Also the precision on the combination of personal ratings (e.g. website) and geotemporal fit is not very high. However, as the neutral items are actually interpreted as bad suggestions, this measure seems to be quite conservative.

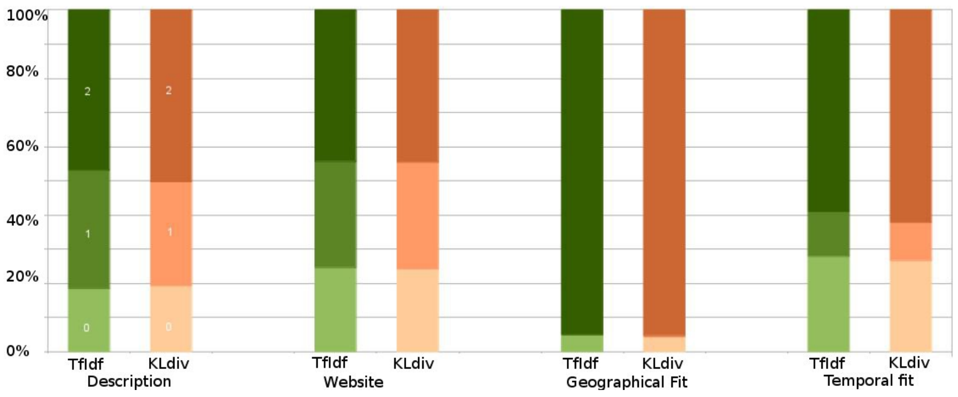

Figure 1: Accuracies on the ratings and contextual fits (0, 1 and 2)

A more detailed look on the distribution of positive, neutral and negative ratings is given in figure 1. The two left-most columns of figure 1 show that approximately half of the suggestions are perceived as interesting (rating 2) when it comes to the opinion of the users. Many items (a third) are perceived as neutral (rating 1). This may mean that the user is not yet sure if he/she would want to follow up on the suggestion, in any case the user is not negative on the suggestion. Overall, around 80% (the sum of the 1 and 2 ratings) of the suggestions are perceived as positive when only the description is shown. When the website is shown the users are a little less positive.

The two right-most columns of figure 1 show that there is a big difference between the accuracy of the suggestions in terms of the geographical fit to the context and the temporal fit to the context. The difference between the TFIDF measure and the Kullback-leibler divergence measure is again neglectable. 95% of the suggestions fit the geographical context. We were surprised that still almost 5% of the suggestions did not match the geographical context, because we used the longitude and latitude as input in Google Places. After manual examination of the ratings we found that there were several errors in the ratings of the geographical fit. We checked 27 suggestions that were judged as a mis-fit, and only 4 of them did indeed not match the geographical context. Thus it seems likely that the actual accuracy in terms of geographical fit is higher. The temporal context is matched in 62% of the suggestions. This leaves room for improvement. After inspection we noticed that theatres and night clubs tend to be suggested during the day as well. This is caused by the opening hours of the box office, which are usually in the afternoon and thus according to our algorithm a suitable suggestion for the afternoon context.

5 Discussion

In the end we struggled with the number of recommendations that we needed to give. We were expected to present each user with 50 suggestions for each context. Although we collected 70-100 search results per context, there were many examples that were filtered out because they did not fit the temporal context. This means that for 34 of the 50 contexts, we did not provide enough suggestions. On average we suggested 44,7 locations for a context. However, since only the top 5 suggestions were evaluated this has no effect on our results.

We noticed that there is not much variation between suggestions for one person and the other. This was partly because of the limited example data. In the training data, each person rated the same items, and the users were generally very positive about most examples. This meant that the similarity between profiles was high. There was also little variation in the results between the Kullback-Leibler method and the TFIDF with cosine similarity method. This is partly because the two method overlap in 4 out of 5 steps; they only varied in the initial rating and even then, both are a measure of term importance within the "positive" profiles.

We also wonder whether personal characteristics are very important in suggesting touristic activities. After all, people often go to the main points of interest when they visit a city anyway. It is important that these are part of the suggestions. Especially when only looking at the top-5 suggestions, it is likely, and maybe also preferred, that these are formed of the important sights of a certain location.

Additionally, the training examples were places from the area of residence of the users. It is possible that when rating places that you are familiar with, you have other preferences then when it comes to places that you have not visited before.

We think we have several strong points in our approach. Overall it is attractive that our approach is completely automated. We think we suggest search results that match the contexts well. This is because we used search results from Google Places, which allowed us to use precise location information in the search query and provided us with opening hours in the results. It is confirmed by the results on the precision of the geographical dimension. However, the information on opening hours was not available for all search results. For some club or theatre locations the opening hours of the box office were used rather than those of the club or theatre itself. Additionally, it was difficult to determine a correct fit to the temporal context automatically. This was caused by the categorical nature of the context (i.e. morning, afternoon or evening). For example should a place that is open until 8 p.m. be suggested for the temporal context evening? The results confirmed that the temporal fit of suggestions need to be improved.

Secondly, we think it is attractive to mix several ranking methods. This way we could find a balance between personalized suggestions and more generic famous places suggestion. Additionally, we could use the opinion of people that have visited the sight already.

And finally, we think the use of reviews as a description for the search result is attractive, since it gives a personal touch to the suggestion even though the descriptions are not personalized. A positive review may influence people, making them more prone to rate the result positively. Overall, people responded a little better to our descriptions than to the website (see table 1).

6 Acknowledgements

This publication was supported by the Dutch national program COMMIT (project P7 SWELL).

References

[1] C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.*, 19(1):1–27, Jan. 2001.

[2] S. Kullback and R. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[3] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.