IRIT at TREC 2012 Contextual Suggestion Track

Gilles Hubert Guillaume Cabanac

Department of Computer Science, University of Toulouse, IRIT UMR 5505 CNRS
118 route de Narbonne, F-31062 Toulouse cedex 9, France
{gilles.hubert, guillaume.cabanac}@univ-tlse3.fr

Abstract. In this paper we give an overview of the participation of the IRIT lab, University of Toulouse, France, in the TREC 2012 Contextual Suggestion Track. We present our personalized contextual retrieval framework, an approach for context processing, and two approaches for user preference processing. The official evaluations of our two submitted runs are reported and compared to the four baselines defined for the track. Evaluation results show that one of our submitted run was ranked 1st among the 27 runs of the 14 participants for the two official evaluation measures of the track.

1 Introduction

In TREC 2012, the Contextual Suggestion Track was run for the first time. The track goal was to investigate search techniques considering context elements such as the user's location, weather, and time, as well as considering user interests via personal preferences and past history, for example. The original aspect of this new track is to tackle simultaneously two issues: 1) searching places according to a given spatial and temporal context and 2) personalizing search results according to user interests.

This track was an opportunity for us to extend previous works on user profiles [BHM02, HM07], geographic information retrieval [PCSH10a, PCSH10b, PSC$^+$12], and information retrieval [Hub05, HM09].

2 Personalized Contextual Retrieval Framework

Our retrieval framework is based on a modular architecture as illustrated in Figure 1. It combines two modules: the first module is dedicated to context processing while the second module is dedicated to preference processing, i.e., to result personalization according to user interests.

The goal of the Contextual Suggestion Track 2012 was to recommend interesting places and activities from the open web. Rather than building an *ad hoc* web search engine we decided to establish our framework on the Google Places (`https://developers.google.com/places`) search engine. This choice is motivated by the fact that the Google Places API retrieves a list of places matching any query with geographic coordinates and searched place types. In addition, we defined several processes to process 1) the input data available to the track and 2) the data returned by the external search engines that we relied on. These processes are related to:

- The definition of queries submitted to the Google Places search engine from the contexts given for the track and the "accepted place types."

- The definition of search result descriptions; our framework uses the Bing (`http://www.bing.com`) and Google (`http://www.google.com`) search engines to collect snippets. These helped to build descriptions about the places returned by the Google Places search engine.
- The modeling of user preferences from the examples and profiles given for the track.
- The personalization of search results according to user preferences.

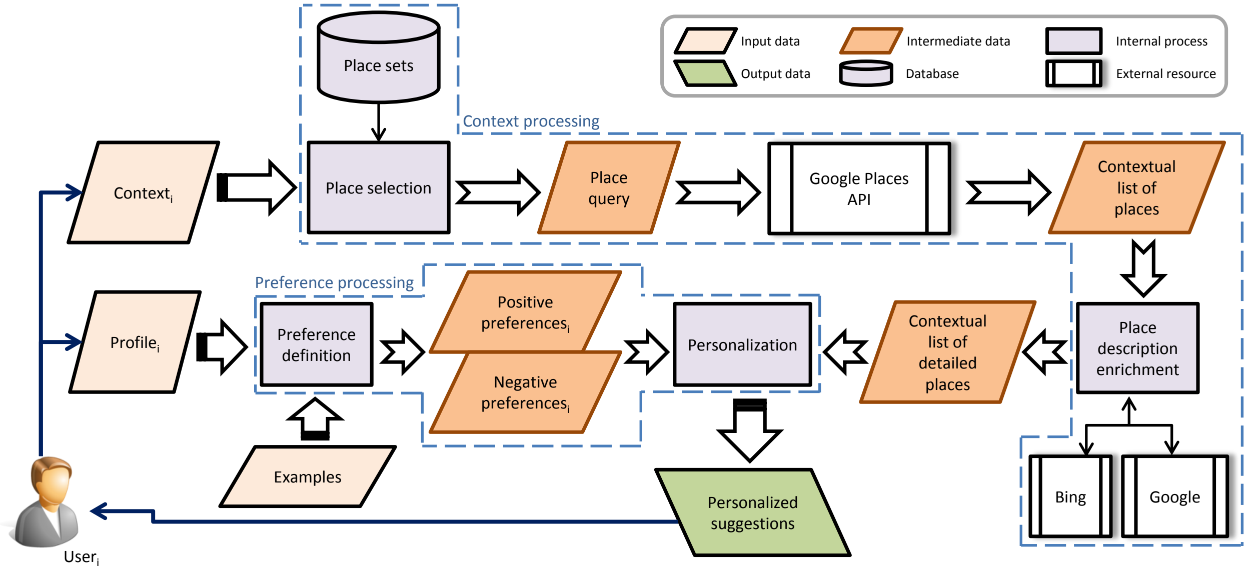

Figure 1: Architecture of our personalized contextual retrieval framework

3 Context Processing

Since our retrieval framework relies on the Google Places API to get a set of places matching a given context, our module for context processing mainly consists in defining a query suited to this external search engine and corresponding to the context. Among the different tags constituting the contexts proposed in the track, we used the following ones: lat (i.e., latitude), long (i.e., longitude), day, time, and season. Geographic coordinates were directly processed by the Google Places API. We defined different sets of place types based on the list of place types accepted by the Google Places API. These place sets were defined intuitively to match the different combinations of temporal parts of the contexts of the track. The different place sets that we came up to were:

- PS 1: amusement_park, aquarium, art_gallery, bar, book_store, bowling_alley, cafe, movie_theater, museum, park, restaurant, shopping_mall, zoo.
- PS 2: aquarium, art_gallery, bar, book_store, bowling_alley, cafe, movie_theater, museum, park, restaurant, shopping_mall.
- PS 3: bar, cafe, grocery_or_supermarket.
- PS 4: bar, cafe, restaurant, shopping_mall.
- PS 5: bowling_alley, cafe, casino, movie_theater, night_club, park, restaurant.
- PS 6: bowling_alley, cafe, casino, movie_theater, night_club, restaurant.

For each context C, a place set was selected depending on the season, the day, and the time given. The association between each triple (season, time, day) and a place type set was achieved manually, as reported in Table 1.

Table 1: Place type sets per triple Season–Time–Day

Season	Time					
	Morning		Afternoon		Evening	
	weekday	weekend	weekday	weekend	weekday	weekend
Spring	PS 3	PS 3	PS 4	PS 1	PS 5	PS 5
Summer	PS 3	PS 3	PS 4	PS 1	PS 5	PS 5
Fall	PS 3	PS 3	PS 4	PS 1	PS 6	PS 6
Winter	PS 3	PS 3	PS 4	PS 2	PS 6	PS 6

Due to a limitation in the Google Places API output to 60 results per query, we split each query into 3 subqueries (i.e., 3 place type subsets with the same geographic coordinates) to retrieve more results (i.e., 3×60). The results returned for the three subqueries were then merged into a single result list. Considering the requirements of the track, we selected only results comprising a website URL to build each list of results corresponding to each context. The result list was then enriched with descriptions of results. For each result, we constituted a description by searching for and merging snippets from Bing (or Google when Bing failed to retrieve any result) based on its website URL.

4 Preference Processing

In our framework, user preferences were processed after searching for the results according to contextual elements (cf. section 3). The list of results L obtained for any context C was personalized for each profile P of the TREC Contextual Suggestion Track.

Tying up with principles of content-based filtering [BC92] we built user preferences from ratings on the examples provided by the track (i.e., the track profiles) and the example contents. User preferences were classified into positive preferences and negative preferences [HMIH00].

Ratings on example suggestions defining the track profiles were classified into initial ratings (i.e., based on the title and description of each example suggestion only) and final ratings (i.e., based on the content of the page and linked pages). For this year's participation, we considered positive ratings to build positive preferences and negative ratings to build negative preferences. To compare positive and negative preferences to contextual results we constructed compatible representations relying on the Vector Space Model [Sal79] widely used in information retrieval (IR). Based on previous works in IR [Hub05, HM09] we performed an indexing process of profiles and contextual results by removing stopwords and keeping representative terms without stemming. We then computed a similarity score between each result vector and each preference vector based on the cosine measure.

We defined two approaches to build and use positive and negative preferences:

1. The coarse-grained approach consisted in defining positive and negative global preferences. For each user profile, the positively rated example suggestions were indexed and merged to constitute a unique vector of terms representing the positive preferences of each user. In the same way, negatively rated suggestions were indexed and merged to constitute a unique vector of terms representing the negative preferences of each user. For each user profile P and each

result r in L, the similarity score was then computed as follows:

$$score_{cg}(P,r) = cosine(Pref^{+}(P),r) - cosine(Pref^{-}(P),r), \qquad score \in [-1,1] \quad (1)$$

This method intended to promote the results most similar to global positive preferences and most dissimilar to global negative preferences.

2. The fine-grained approach consisted in defining positive and negative preferences as sets of positive and negative preference examples. For each user profile, the positively rated example suggestions were indexed to constitute a set of term vectors representing the positive preferences of each user (E^{+}). In the same way, negatively rated suggestions were indexed to constitute a set of term vectors representing the negative preferences of each user (E^{-}). For each user profile P and each result r, the similarity score was then computed as follows:

$$score_{fg}(P,r) = \max_{l \in E^{+}}(cosine(Pref_{l}^{+}(P),r)) - \max_{m \in E^{-}}(cosine(Pref_{m}^{-}(P),r)), score \in [-1,1] \quad (2)$$

This method intended to promote the results most similar to one positive example and most dissimilar to one negative example.

Results were then ordered by decreasing order of scores.

5 Runs and Evaluation Results

We discuss in this section the features as well as the official evaluations of the two runs that we submitted to the TREC 2012 Contextual Suggestion Track and the features of the four baselines created by the organizers of the track.

5.1 Submitted Runs

For this year's participation to the TREC Contextual Suggestion Track we submitted two runs labeled *iritSplit3CPv1* and *iritSplit3CPv2* based on both context and preference processing. We applied the following same principles for both runs:

- According to the submission guidelines, we selected the top 50 results from the result lists returned by our framework to populate the run.
- For this first participation, positive preferences were based on examples with initial and final ratings equal to 1, considering only examples of great interest for the users with regards to either their description or page contents. We indexed only description parts without considering the content of the page and linked pages. In a similar way, negative preferences were based on examples with initial and final ratings equal to -1.

The two runs differed from the approach applied for preference processing (cf. section 4):

- The first run *iritSplit3CPv1* applied the coarse-grained approach of preference processing that uses global positive and negative user preferences.
- The second run *iritSplit3CPv2* applied the fine-grained approach of preference processing that uses positive and negative preferences as sets of positive and negative preference examples.

5.2 Baselines

Four baselines were defined by the organizers of the track as follows:

- The baseline *waterloo12a* consisted of the top 50 attractions given by tripadvisor.com for the city of each context. This baseline took into account the geographical aspect only, thus ignoring the user profiles and temporal aspect of information needs.
- The baseline *waterloo12b* also consisted of the top 50 attractions given by tripadvisor.com. Contrary to *waterloo12a* it used the site's search tool with search terms manually generated based on the sample suggestions with a high rating in the user's profile.
- The baseline *baselineA* was output from the Google Places API. Each context was split into day, time, and location. The suggestions were taken from the top-rated results of the union of query results covering the context and sorted by Google's own ratings. This baseline did not rely on the use of user profiles.
- The baseline *baselineB* was created the same way as *baselineA* but suggestions were restricted to pubs, restaurants, and cafes.

5.3 Results

The evaluation results according to the P@5 measure over profiles and contexts for the different dimensions (i.e., W: Website, G: Geographical, T: Temporal, and D: Description) and combinations (i.e., WGT and GT) of these two runs and the four baselines are reported in Table 2.

Table 2: Official results for the P@5 measure

Run	P@5 (Mean over all the profiles and contexts)					
	WGT	GT	G	T	W	D
baselineA (+)	0.1784	0.5114	0.7908	0.5694	0.4086	0.3031
baselineB	0.1704	0.5482	0.8060	0.5883	0.2654	0.2444
waterloo12a	0.1377	0.4229	0.8230	0.4451	0.3463	0.3272
waterloo12b (−)	0.0864	0.4065	0.6827	0.4988	0.1741	0.3117
iritSplit3CPv1	**0.3235**	**0.6027**	**0.8930**	**0.6156**	**0.4599**	**0.3605**
iritSplit3CPv2	0.1790	0.5486	0.8466	0.5580	0.3235	0.2593

According to Table 2, *baselineA* is a strong baseline for the official P@5 WGT measure (it is higher than the other baselines). Our runs outperformed this strong baseline and therefore all the four baselines. Our *iritSplit3CPv1* run particularly yielded better results than the strong baseline according to all the dimensions and dimension combinations. In particular, it was largely higher (+81.33 %) than the strong baseline according to the official P@5 WGT measure as showed in Table 3. This comparison is interesting since *iritSplit3CPv1* and *baselineA* used the same external search engine. It shows that the modules we defined for context processing to result personalization provided a material improvement in effectiveness.

The evaluation results according to the MRR measure over profiles and contexts for the different dimensions and their combinations of these two runs and the four baselines are reported in Table 4.

According to Table 4, *baselineB* serves as strong baseline for the official MRR WGT measure, instead of *baselineA*. Our *iritSplit3CPv1* run outperformed this strong baseline according to all the dimensions and dimension combinations, except for the temporal dimension. In particular, this run yielded greater effectiveness (+33.42 %) compared to the strong baseline according to the

Table 3: Difference with the strong baseline for the P@5 measure

Difference	P@5 (Mean over all the profiles and contexts)					
iritSplit3CPv1	**WGT**	GT	G	T	W	D
vs BaselineA (+)	+81.33 %	+17.85 %	+12.92 %	+8.11 %	+12.56 %	+18.94 %

Table 4: Official results for the MRR measure

Run	MRR (Mean over all the profiles and contexts)					
	WGT	GT	G	T	W	D
baselineA	0.2993	0.6447	0.8906	0.7002	0.5366	0.4632
baselineB (+)	0.3504	0.7470	0.9274	**0.7817**	0.4384	0.3951
waterloo12a	0.2130	0.5703	0.8615	0.6119	0.3859	0.4183
waterloo12b (−)	0.1404	0.5304	0.7447	0.6149	0.2775	0.4467
iritSplit3CPv1	**0.4675**	**0.7585**	**0.9480**	0.7634	**0.6493**	**0.5461**
iritSplit3CPv2	0.3377	0.6795	0.9072	0.6853	0.4500	0.3841

official MRR WGT measure as showed in Table 5. Once again it is worth noting that *iritSplit3CPv1* and *baselineB* used the same external search engine, which gives credit to our context processing and result personalization. In contrast, our second run (*iritSplit3CPv2*) yielded poorer evaluation results, except for the website dimension.

Table 5: Difference with the strong baseline for the MRR measure

Difference	MRR (Mean over all the profiles and contexts)					
iritSplit3CPv1	**WGT**	GT	G	T	W	D
vs BaselineB (+)	+33.42 %	+1.54 %	+2.22 %	−2.34 %	+48.11 %	+38.22 %

On the one hand, evaluations reported in Table 2 and Table 4 show that our runs (*iritSplit3CPv1* and *iritSplit3CPv2*) yielded better evaluations for the Geographical and Temporal dimensions than for the Website and Description dimensions. The poorer results for the Description dimension could be explained by the raw fusion of snippets that we applied to create suggestion descriptions.

On the other hand, evaluations show that *iritSplit3CPv1* overcomes *iritSplit3CPv2* regarding all the dimensions and dimension combinations. The superiority of *iritSplit3CPv1* is particularly visible with regards to the Website and Description dimensions and the WGT combination (i.e., combination of the Website, Geographical, and Temporal dimensions). These results show that our coarse-grained approach returned more suitable suggestions than our fine-grained approach for preference processing. An explanation for that could be that the fine-grained approach alloted too much importance to two extreme examples only (i.e., the best matching positive example and the best matching negative example) leaving other examples aside.

In addition, our two runs obtained promising results compared to the statistics over the 27 submitted runs. *iritSplit3CPv1* was ranked 1st according to the two official measures (i.e., P@5 WGT and MRR WGT). *iritSplit3CPv2* was ranked 14th according to P@5 WGT and 12th according to MRR WGT. Moreover, according to the WGT combination, *iritSplit3CPv1* was greater or equal to the Median for 34/35 judged contexts over profiles and 8 times the best run. It was also greater or equal to the Median for 17/19 judged profiles over contexts and 6 times the best run. *iritSplit3CPv2* was greater or equal to the Median for 30/35 judged contexts over profiles and 4 times the best run. It was also greater or equal to the Median for 15/19 judged profiles over contexts and 2 times the best run.

6 Conclusion and Future Work

This paper introduced our personalized contextual framework based on a modular architecture combining two modules. A first module is dedicated to context processing. A second module is dedicated to result personalization according to user interests. This year's participation to the TREC Contextual Suggestion Track led to encouraging results since one of our submitted runs was ranked 1st according to the two official evaluation measures. We checked that the effectiveness of our approach does not only depend on the data that we extracted from Google Places, since our run outperforms baselines that rely on these data by a margin of 81.33 % or more (for the official P@5 WGT measure).

Future work will tackle the temporal processing of queries (i.e., season, time, and day) to associate place type sets to contexts with higher accuracy. We will also tackle preference processing with regards to partial positive and negative ratings and with regards to similarity scoring between contextual results and user preferences.

References

[BC92] Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35(12):29–38, 1992.

[BHM02] Anis Benammar, Gilles Hubert, and Josiane Mothe. Automatic profile reformulation using a local document analysis. In *Advances in Information Retrieval, 24th BCS-IRSG European Colloquium on IR Research*, volume 2291 of *LNCS*, pages 124–134. Springer, 2002.

[HM07] Gilles Hubert and Josiane Mothe. Reusing past queries to facilitate information retrieval. In *ICSOFT '07: Proceedings of the 2nd International Conference on Software and Data Technologies*, volume 3, pages 166–171. INSTICC Press, 2007.

[HM09] Gilles Hubert and Josiane Mothe. An adaptable search engine for multimodal information retrieval. *J. Am. Soc. Inf. Sci. Technol.*, 60(8):1625–1634, 2009.

[HMIH00] Keiichiro Hoashi, Kazunori Matsumoto, Naomi Inoue, and Kazuo Hashimoto. Document filtering method using non-relevant information profile. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 176–183, New York, NY, USA, 2000. ACM.

[Hub05] Gilles Hubert. A voting method for XML retrieval. In *Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Revised Selected Papers*, volume 3493 of *LNCS*, pages 183–196. Springer, 2005.

[PCSH10a] Damien Palacio, Guillaume Cabanac, Christian Sallaberry, and Gilles Hubert. Measuring Effectiveness of Geographic IR Systems in Digital Libraries: Evaluation Framework and Case Study. In *ECDL'10: Proceedings of the 14th European Conference on Research and Advanced Technology for Digital Libraries*, volume 6273 of *LNCS*, pages 340–351. Springer, 2010.

[PCSH10b] Damien Palacio, Guillaume Cabanac, Christian Sallaberry, and Gilles Hubert. On the evaluation of geographic information retrieval systems: Evaluation framework and case study. *Int. J. Digit. Libr.*, pages 91–109, 2010.

[PSC⁺12] Damien Palacio, Christian Sallaberry, Guillaume Cabanac, Gilles Hubert, and Mauro Gaio. Do expressive geographic queries lead to improvement in retrieval effectiveness? In *Bridging the Geographic Information Sciences, International AGILE'2012 Conference*, Lecture Notes in Geoinformation and Cartography, pages 267–286. Springer Berlin Heidelberg, 2012.

[Sal79] Gerard Salton. Mathematics and information retrieval. *J. Doc.*, 35(1):1–29, 1979.