

IRIT, GeoComp, and LIUPPA at the TREC 2013 Contextual Suggestion Track

Gilles Hubert¹, Guillaume Cabanac¹, Karen Pinel-Sauvagnat¹,
Damien Palacio², and Christian Sallaberry³

¹Department of Computer Science, University of Toulouse, IRIT, France*

²Department of Geography, University of Zurich, Switzerland

³LIUPPA, University of Pau et des Pays de l'Adour, France

Abstract. In this paper we give an overview of the participation of the IRIT, GeoComp, and LIUPPA labs in the TREC 2013 Contextual Suggestion Track. Our framework combines existing geo-tools or services (e.g., Google Places, Yahoo! BOSS Geo Services, PostGIS, Gisgraphy, GeoNames) and ranks results according to features such as context-place distance, place popularity, and user preferences. We participated in the Open Web and ClueWeb12 sub-tracks with runs `IRIT.OpenWeb` and `IRIT.ClueWeb`.

1 Introduction

Our participation to TREC 2013 Contextual Suggestion Track allowed us to revise our framework experimented in the previous edition of the track [HC12]. The 2012 framework has some limitations that we intended to overcome, such as checking result contents (i.e., webpages) to exclude inadequate pages (e.g., general information not related to a given location) and providing more accurate descriptions about results according to the user's context (orientation, distance in meter/time, other interesting POIs around, and so on). The new framework we propose triggers additional geo-tools and services, as well as description enriching, filtering and ranking processes based on all collected data (e.g., preferences, popularity, and proximity). In addition, the new framework, first designed to search the web, was then adapted and experimented on the ClueWeb12 sub-collection provided this year. We submitted two runs, one run per sub-track: `IRIT.OpenWeb` and `IRIT.ClueWeb`. The process flows for both runs is composed of four main stages, described in Figure 1.

2 Run `IRIT.OpenWeb` on the Open Web

2.1 Preference Processing

We identified two ways for processing user preferences (Figure 1a, process 1). In parallel, we (i) build positive and negative term preferences for the user, and (ii) we define his/her categories of interest.

Positive and negative term preferences were extracted from description ratings only. We considered ratings with values 3 and 4 to build positive preferences and ratings with values 0 and 1 to build negative preferences. For each preference, we indexed the corresponding descriptions of the selected examples by removing stopwords and keeping representative terms without stemming to build a term vector [Sal79].

Categories of user interests either correspond to Google Places categories (GP)¹ or to our own defined categories (OD), based on Swarbrooke's categorization [Swa95]. To associate users with categories of interest, we first automatically labeled each place (of the example suggestions) with one or more categories. An example is labeled with a category and an associated confidence score if this example is retrieved by Terrier using the category as a query (for our own categories, we also used the hyponyms of the category returned

*gilles.hubert@univ-tlse3.fr

¹https://developers.google.com/places/documentation/supported_types

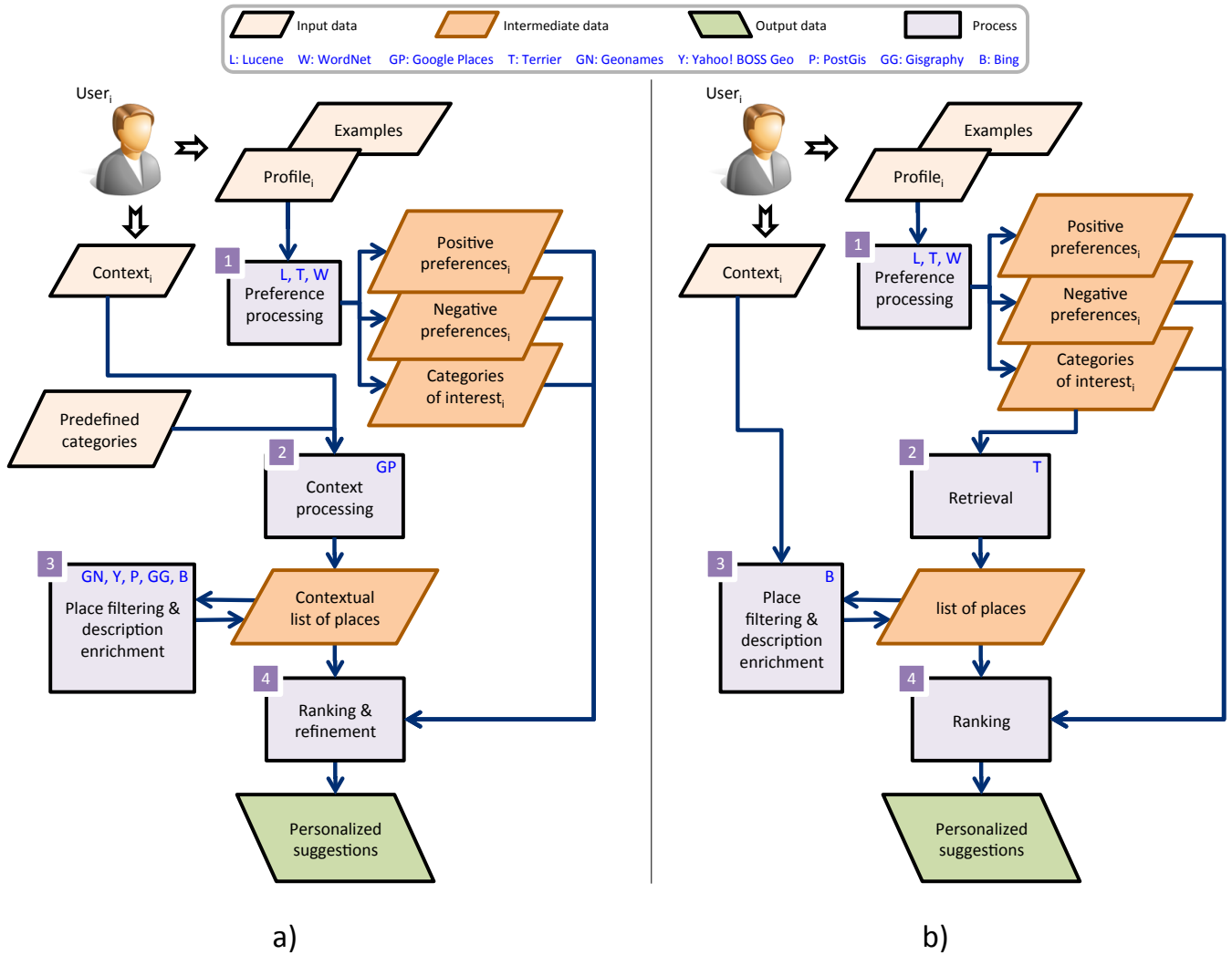


Figure 1: Architecture of our personalized contextual retrieval process

by WordNet as queries). The confidence score corresponds to the retrieval score of Terrier. Then, we defined the categories of interest of a user as the categories corresponding to the examples for which he/she gave a positive vote regarding the description. A confidence score $w(c, u)$ is also associated with the user's categories of interest. It is evaluated as $w(c, u) = \sum_{e \in \{examples\}} ((ratingDesc(u, e) + ratingUrl(u, e)) * w(c, e))$, where $ratingDesc(u, e)$ is the rating of user u on the description of example e , $ratingUrl(u, e)$ is the rating of user u on the URL of example e , and $w(c, e)$ is the weight of the category c for example e (i.e., the confidence score indicating the level of representativeness of category c in example e).

2.2 Context Processing

As showed in Figure 1a (process 2), for each context we first queried Google Places with 1) its GPS coordinates and 2) all aforementioned categories as keyword. The result list ranks places according to their proximity from the context. Each place is described by its name, address, location (GPS coordinates), website URL, rating extracted from social media, and the initial categories that matched the place. Note that this process does not consider any characteristics of the users.

2.3 Place Filtering and Enrichment

This process (Figure 1a, process 3) consists of two phases.

First, it filters inadequate pages (URLs) with the Yahoo! BOSS Geo Services geo-parsing tools. A page is discarded when no relevant placenames are detected within its text. A page may have no placenames or it may be too far away from the user’s context. We used Google MatrixDistance API to compute such a car distance in minutes. Considering all user contexts, overly frequent URLs were also blacklisted, since it is likely that they refer to global corporations (e.g., www.mcdonalds.com) vs. local places.

Second, the descriptions of remaining places were enriched. We used distance computing results (Google MatrixDistance) to get distance by feet and car distance between the user location and that of the retrieved places. Other geo-tools (e.g., PostGIS) were used to calculate Euclidian distance and orientation information between the user location and the retrieved places. Gisgraphy, based on GeoNames, processes URLs and generates a set of nearby POIs for each place. Gisgraphy suggests categories to stamp on the POIs (e.g. restaurant, park. . .). Finally, we used Microsoft Bing search engine to get short descriptions of URLs.

2.4 Suggestion Ranking and Refinement

The final step of our approach is to return for each pair (user, context) a ranked list of documents. This list stems from two sets of candidate documents, computed based on evidences from GP on the one hand, and evidences from OD on the other hand. Each document is associated with 6 scores:

1. $s_1(p, u)$ reflects the matching between the categories of place p and user u categories inferred as explained in section 2.1. Places that did not match any of the user categories were first dismissed. Finally $s_1(p, u) = \sum_{i \in C} w(c_i, u)$ for all remaining categories $c_i \in C$.
2. $s_2(p, u) = |C|$ is the number of categories in common between user categories of interest and the categories of the considered place.
3. $s_3(p)$ is the number of POIs located around the place p , evaluated with GeoNames.
4. $s_4(p, u)$ is the Euclidian distance between place p and the location of user u .
5. $s_5(p)$ is the rating value provided by Google Places about place p . Places without rating were ranked at the bottom of the list.
6. $s_6(p, u) = ps(p, u) - ns(p, u)$ is the similarity between u and p based on full-text of the document describing p . The positive score ps is the cosine value between the vector of user positive preferences and the document vector. ns is computed likewise with negative preferences.

These 6 scores are then transformed to be uniformly distributed and defined on the same domain of values (range). The final score of a place given a user and a context is then computed. Places present in the two sets (GP and OD) are ranked higher in the final list. This is done by summing up the scores of the document in the two sets. These 12 scores are summed altogether with the following weighting scheme: $1/3$ for s_1 , $1/3$ for s_6 , and $1/12$ for the remaining factors.

3 Run IRIT.ClueWeb on the ClueWeb12 Sub-collection

For this sub-track, we indexed the ClueWeb12 sub-collection with Terrier, without considering the different contexts. We also identified categories of interest for each user, as done for the Open Web sub-track (see section 2.1). These categories of interest either correspond to Google Places categories (GP) or to our own categories (OD). We then submitted two queries to the Terrier search engine for each user: they were respectively composed of his/her weighted GP/OD categories of interest.

We finally applied the ranking method based on ranks used on the open web with three factors: i) the RSV of Terrier with the GP query, ii) the RSV of Terrier with the OD query, and iii) the s_6 score, computed with the Lucene search engine. The resulting ranked list is user-dependent but context independent. For a given context, we thus tailored the list by removing the documents irrelevant to the context at stake. Each place suggestion was then returned associated with a description featuring the snippet returned by Bing using the URL of the document.

4 Evaluation Results

Figure 2 shows the evaluation results for the run labeled `IRIT.OpenWeb` according to the $P@5$ measure. The x -axis shows groups of topics, provided that a topic is a pair (profile, context). Five groups of topics are constituted according to the best $P@5$ reached by participants. The difficulty of topics ranges from “easy” to “impossible” since no participants suggested any relevant place among the top 5 documents. The y -axis shows the number of topics having $P@5 = x$. For instance, our system retrieved 24 easy topics with a $P@5 = 0.60$. This corresponds to 29 % of the easy topics, as showed over the third bar.

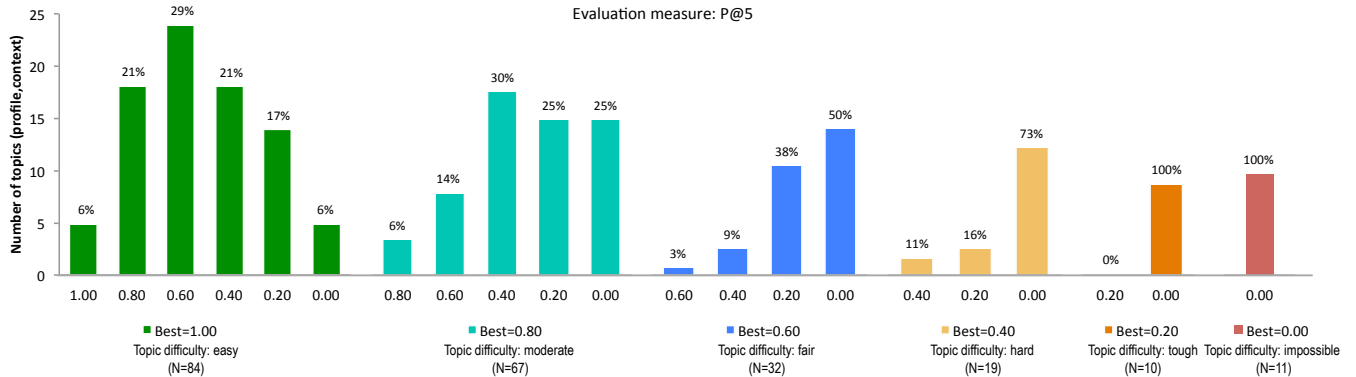


Figure 2: Analysis of our results for the run `IRIT.OpenWeb` according to $P@5$

Evaluation results according to the $P@5$ measure for the run labeled `IRIT.ClueWeb` are more contrasted since 27 % of the topics comprised relevant documents among the top 5 documents.

Table 1 shows the official results of our runs compared with the baselines of the two sub-tracks.

Run	$P@5$	MRR	TBG
<code>IRIT.OpenWeb</code>	0.3112	0.4915	1.4638
BaselineA	0.1372	0.2316	0.5234
<code>IRIT.ClueWeb</code>	0.0798	0.1346	0.3279
BaselineB	0.1417	0.2452	0.4797

The `IRIT.OpenWeb` run outperforms significantly the BaselineA run that relies on Google Places. This means that the additional processes used in our approach leads to effectiveness improvement. Other experiments are however needed to evaluate each part of our framework. Regarding the `IRIT.ClueWeb` run, we considered that all documents in the collection were still accessible by assessors, which was unfortunately not always the case. This can explain the poor results of our run compared to the BaselineB run.

5 Conclusion and Future Work

This article introduced our personalized contextual framework, with a focus on the evolutions integrated since the previous TREC Contextual Suggestion Track in 2012. Evolutions intended to overcome previous limitations regarding inadequate suggestions (e.g., general information not related to a given location) provided to users and poorly informative descriptions of suggestions. We integrated additional geo-tools and services (such as Yahoo! BOSS Geo Services, PostGIS, Gisgraphy, and GeoNames), as well as filtering and ranking processes based on all the available information (e.g., preferences, popularity, proximity). The framework was also adapted to process the ClueWeb12 Contextual Suggestion sub-collection provided this year.

Future work will tackle ranking algorithms giving a weight distributed on the different arguments involved. In addition, we plan to test our framework with the whole ClueWeb12 collection.

References

- [HC12] Gilles Hubert and Guillaume Cabanac. IRIT at TREC 2012 Contextual Suggestion Track. In *Proceedings of the 21th Text REtrieval Conference*, Gaithersburg, MD, USA, 2012. <http://trec.nist.gov/pubs/trec21/papers/IRIT.context.final.pdf>.
- [Sal79] Gerard Salton. Mathematics and information retrieval. *Journal of Documentation*, 35(1):1–29, 1979.
- [Swa95] John Swarbrooke. *The Development and Management of Visitor Attractions*. Oxford: Butterworth Heinemann, 1995.