On Suggesting Entities as Web Search Queries

Extended Abstract

Diego Ceccarelli[1,2,3], Sergiu Gordea[4], Claudio Lucchese[1], Franco Maria Nardini[1], and Raffale Perego[1]

[1] ISTI-CNR, Pisa, Italy – {firstname.lastname}@isti.cnr.it
[2] IMT Institute for Advanced Studies Lucca, Lucca, Italy
[3] Dipartimento di Informatica, Università di Pisa, Pisa, Italy
[4] AIT GmbH, Wien, Austria – sergiu.gordea@ait.ac.at

Abstract. The Web of Data is growing in popularity and dimension, and named entity exploitation is gaining importance in many research fields. In this paper, we explore the use of entities that can be extracted from a query log to enhance query recommendation. In particular, we extend a state-of-the-art recommendation algorithm to take into account the semantic information associated with submitted queries. Our novel method generates highly related and diversified suggestions that we assess by means of a new evaluation technique. The manually annotated dataset used for performance comparisons has been made available to the research community to favor the repeatability of experiments.

1 Semantic Query Recommendation

Mining the past interactions of users with the search system recorded in query logs is an effective approach to produce relevant query suggestions. This is based on the assumption that *information* searched by past users can be of interest to others. The typical interaction of a user with a Web search engine consists in *translating her information need in a textual query made of few terms.* We believe that the "*Web of Data*" can be profitably exploited to make this process more user-friendly and alleviate possible vocabulary mismatch problems.
We adopt the *Search Shortcuts* (SS) model proposed in [1, 2]. The SS algorithm aims to generate suggestions containing only those queries appearing as final in successful sessions. The goal is to suggest queries having a high potentiality of being useful for people to reach their initial goal. The SS algorithm works by efficiently computing similarities between partial user sessions (the one currently performed) and historical successful sessions recorded in a query log. Final queries of most similar successful sessions are suggested to users as **search shortcuts**.

A virtual document is constructed by merging successful session, i.e., ending with a clicked query. We annotate virtual documents to extract relevant named entities. Common annotation approaches on query logs consider a single query and try to map it to an entity (if any). If a query is ambiguous, the risk is to always map it to the most popular entity. On the other hand, in case of

ambiguity, we can select the entity with the highest likelihood of representing the semantic context of a query.

We define *Semantic Search Shortcuts* (S^3) the query recommender system exploiting this additional knowledge. Please note that S^3 provides a list of related entities, differently from traditional query recommenders as SS that for a given query produce a flat list of recommendations. We assert that entities can potentially deliver to users much more information than raw queries.

In order to compute the entities to be suggested, given an input query q, we first retrieve the top-k most relevant virtual documents by processing the query over the SS inverted index built as described above. The result set R_q contains the top-k relevant virtual documents along with the entities associated with them. Given an entity e in the result set, we define two measures:

$$score(e, \mathcal{VD}) = \begin{cases} conf(e) \times score(\mathcal{VD}), & \text{if } e \in \mathcal{VD}.entities \\ 0 & \text{otherwise} \end{cases}$$

$$score(e, q) = \sum_{\mathcal{VD} \in R_q} score(e, \mathcal{VD})$$

where $conf(e)$ is the confidence of the annotator in mapping the entity e in the virtual document $\mathcal{VD}$, while $score(\mathcal{VD})$ represents the similarity score returned by the information retrieval system. We rank the entities appearing in R_q using their score *w.r.t.* the query.

2 Experimental Evaluation

We used a large query log coming from the Europeana portal[1], containing a sample of users' interactions covering two years (from August 27, 2010 to January, 17, 2012). We preprocessed the entire query log to remove noise (e.g., queries submitted by software robots, mispells, different encodings, etc). Finally, we obtained 139,562 successful sessions. An extensive characterization of the query log can be found in [3]. To assess our methodology we built a dataset consisting of 130 queries split in three disjoint sets: 50 short queries (1 term), 50 medium queries (on average, 4 terms), 30 long terms (on average, 9 terms). For each query in the three sets, we computed the top-10 recommendations produced by the SS query recommender system and we manually mapped them to entities by using a simple interface providing an user-friendly way to associate entities to queries[2].

We are interested in evaluating two aspects of the set of suggestions provided. These are our main research questions:

[1] We acknowledge the Europeana Foundation for providing us the query logs used in our experimentation. http://www.europeana.eu/portal/

[2] Interested readers can download the dataset from: http://hpc.isti.cnr.it/~ceccarelli/doku.php/sss.

Relatedness : How much information related to the original query a set of suggestions is able to provide?

Diversity : How many different aspects of the original query a set of suggestions is able to cover?

To evaluate these aspects, we borrow from the annotators the concept of *semantic relatedness* between two entities proposed by Milne and Witten [4]:

$$rel(e_1, e_2) = 1 - \frac{log(max(|I_L(e_1)|, |I_L(e_2)|)) - log(|I_L(e_1) \cup I_L(e_2)|)}{log(|KB|) - log(min(|I_L(e_1)|, |I_L(e_2)|))}$$

where e_1 and e_2 are the two entities of interest, the function $I_L(e)$ returns the set of all entities that link to the entity e in Wikipedia, and KB is the whole set of entities in the knowledge base. We extend this measure to compute the similarity between two set of entities (the function I_L gets a set of entities and returns all the entities that link *at least* on entity in the given set). At the same time, given two sets of entities E_1, E_2, we define the diversity as $div(E_1, E_2) = 1 - rel(E_1, E_2)$. Given a query q, let E_q be the set of entities that have been manually associated with the query. We define the relatedness and the diversity of a list of suggestions S_q as:

Definition 1 *The average relatedness of a list of suggestions is computed as:*

$$rel(S_q) = \frac{\sum_{s \in S_q} rel(E_s \setminus E_q, E_q)}{|S_q|}$$

where E_s represents the set of entities mapped to a suggestion s (could contain more than one entity in the manual annotated dataset). Please note that we remove the entities of the original query from each set of suggestions as we are not interested in suggesting something that do not add useful content *w.r.t.* the starting query ($E_s \setminus E_q$).

Definition 2 *The average diversity of a list of suggestions is defined as:*

$$div(S_q) = \frac{\sum_{s \in S_q} div(E_s, E_{S_q \setminus s})}{|S_q|}$$

For each suggestion, we intend to evaluate how much information it adds *w.r.t* the other suggestions. $E_{S_q \setminus s}$ denotes the union of the entities belonging to all the suggestions except the current suggestion s.

Experimental Results: For each set of queries in the dataset described above (*short*, *medium* and *long*), we compared the average relatedness and the average diversity of the recommendations generated by SS and by S^3.

Figure 1 shows the average relatedness computed for each query q belonging to a particular set of queries. Results confirm the validity of our intuition as, for all the three sets, the results obtained by S^3 are always better than the results obtained by considering the SS suggestions. It is worth to observe that the longer the queries the more difficult the suggestion of related queries. This

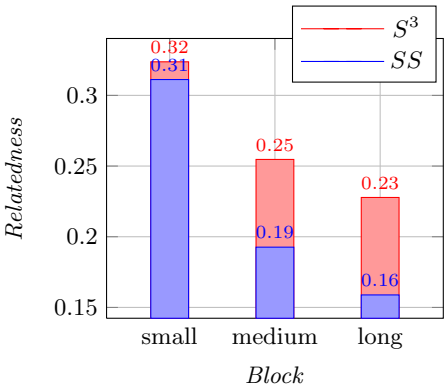

Fig. 1: Per-set average relatedness computed between the list of suggestions and the given query.

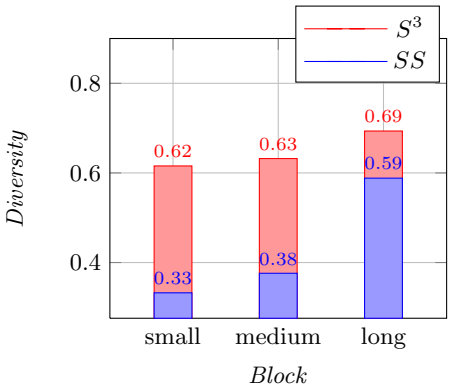

Fig. 2: Per-set average diversity computed between the list of suggestions and the given query.

happens because long queries occur less frequently in the log and then we have less information to generate the suggestions. If we consider single sets, the highest gain of S^3 in terms of average relatedness is obtained for medium and long queries: this means that relying on entities allows to mitigate the sparsity of user data.

Figure 2 reports the average diversity of the suggestions over the queries of each set. Here, we observe an opposite trend, due to the fact that the longer the queries, the more terms/entities they contain, and the more different the suggestions are. Furthermore, we observe that, for the most frequent queries, SS has a very low performance *w.r.t.* S^3. This happens because for frequent queries SS tends to retrieve popular reformulations of the original query, thus not diversifying the returned suggestions. S^3 does not suffer for this problem since it works with entities thus diversifying naturally the list of suggestions. We leave as future work the study of a strategy for suggesting entities aiming at maximizing the diversity on a list of suggestions.

References

1. Baraglia, R., Cacheda, F., Carneiro, V., Fernandez, D., Formoso, V., Perego, R., Silvestri, F.: Search shortcuts: a new approach to the recommendation of queries. In: Proc. RecSys'09. ACM, New York, NY, USA (2009)
2. Broccolo, D., Marcon, L., Nardini, F.M., Perego, R., Silvestri, F.: Generating suggestions for queries in the long tail with an inverted index. IP&M
3. Ceccarelli, D., Gordea, S., Lucchese, C., Nardini, F.M., Tolomei, G.: Improving europeana search experience using query logs. In: Proc. TPDL'11. pp. 384–395
4. Milne, D., Witten, I.: Learning to link with wikipedia. In: Proc. CIKM'08. pp. 509–518. ACM (2008)