Recommendations for the Long Tail by Term-Query Graph

Francesco Bonchi
Yahoo! Research, BCN, Spain
bonchi@yahoo-inc.com

Raffaele Perego
ISTI - CNR, Pisa, Italy
r.perego@isti.cnr.it

Fabrizio Silvestri
ISTI - CNR, Pisa, Italy
f.silvestri@isti.cnr.it

Hossein Vahabi
IMT, Lucca, Italy
hossein.vahabi@imtlucca.it

Rossano Venturini
ISTI - CNR, Pisa, Italy
r.venturini@isti.cnr.it

ABSTRACT

We define a new approach to the query recommendation problem. In particular, our main goal is to design a model enabling the generation of query suggestions also for rare and previously unseen queries. In other words we are targeting queries *in the long tail*. The model is based on a graph having two sets of nodes: *Term* nodes, and *Query* nodes. The graph induces a Markov chain on which a generic random walker starts from a subset of Term nodes, moves along Query nodes, and restarts (with a given probability) only from the same initial subset of Term nodes. Computing the stationary distribution of such a Markov chain is equivalent to extracting the so-called *Center-piece Subgraph* from the graph associated with the Markov chain itself. Given a query, we extract its terms and we set the restart subset to this term set. Therefore, we do not require a query to have been previously observed for the recommending model to be able to generate suggestions.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process, Query formulation

General Terms

Algorithms, Experimentation, Measurement

Keywords

Query Recommender Systems, Web Search Effectiveness

1. BACKGROUND AND RELATED WORK

Query recommendations are an important tool for search engines, as they help users in refining their initial queries towards a better expression of their information needs. Recommendations are typically queries semantically related to the original one, and they are usually obtained by models learned with query logs [5], for instance, by clustering queries [8], or by identifying frequent re-phrasings [1]. Recently, Boldi *et al.* introduced the *Query Flow Graph* [2] (QFG) model, which is a Markov chain-based representation of a query log. A QFG is a directed graph in which nodes are queries, and an edge $\vec{e} = (q_1, q_2)$ exists if at least a user has issued q_2 after q_1. Furthermore, $\vec{e}$ is weighed by the probability of a user to issue q_2 after q_1. Given a query q, suggestions are generated by means of random walks from q on the QFG [3].

All these previous models suffer from the same limitation: they can only provide recommendations for queries present in the training query log. For instance, the QFG model needs query q to be present in the graph in order to start the random walk and provide recommendations. As it was pointed out by Downey *et al.* [4] trough an analysis on search behaviors, rare queries are very important, and their effective satisfaction is very challenging for search engines. Therefore, it is even very important to provide good recommendations for long-tail queries.

A recent work by Yang *et al.* [6] proposes an optimal framework for rare-query suggestion leveraging on implicit feedbacks from users mined from the query logs. However, unknown queries are excluded even in this context. In this paper instead, we introduce a model enabling the generation of query suggestions also for previously unseen queries: to the best of our knowledge this is the first model able to do so. The model is briefly explained next.

2. TERM-QUERY GRAPH MODEL

Let $\overline{Q}$ be a query-log, i.e. a sequence of queries $\langle q_1, \ldots, q_n \rangle$. Moreover, let Q be the set of queries in the query-log. Each query is made up of terms from a vocabulary T, without loss of generality we consider each query q as a subset of the term set T, i.e. $q \subseteq T$. From T and Q we build a *digraph* in the following manner. Each term $t \in T$ and query in $q \in Q$ is a node, we denote by V_T (resp. V_Q) the set of *Term* (resp. *Query*) nodes, i.e. nodes corresponding to terms in T (resp. Q). Likewise, we define two kinds of edges. We denote by E_{TQ} the set of directed edges going from terms in T to queries in Q, $t \to q \in E_{TQ}$ iff $t \in q$. We denote by E_Q the set of direct edges among queries, i.e. $q_1 \to q_2 \in E_Q$ iff the likelihood a user submits q_2 after q_1 is not null. Resuming, we consider a digraph $G = (V_T \cup V_Q, E_{TQ} \cup E_Q)$, where the subgraph $G_Q = (V_Q, E_Q)$ is, basically, the Query-Flow Graph (QFG) [2] built on queries from Q. We call this graph the *Term-Query Graph*, or TQ-Graph for short, and we consider this as the reference structure in our *Term-Query Graph Model*.

Query suggestions for a query $q = \{t_1, \ldots t_m\} \subseteq T$ are generated, within this model, by extracting the *Center-piece Subgraph* starting from the m Term nodes $t_1, \ldots, t_m$ [7]. Therefore, in practice, we compute m random walks with restart from each term in q. Let $\vec{r_{(1)}}, \ldots, \vec{r_{(m)}}$ be the corresponding stationary distributions. Furthermore, let $\vec{r} = \vec{r_{(1)}} \circ \vec{r_{(2)}} \circ \ldots \circ \vec{r_{(m)}}$ be the Hadamard product of the m vectors, i.e. $\vec{r}_i = \prod_{k=1}^{m} \left(\vec{r_{(k)}}\right)_i$. Since each dimension of $\vec{r}$ corresponds to a query in Q, we recommend the queries corresponding to the top scoring ones.

As an example, consider a query-log containing just two queries, q_1, q_2, whose terms are in a vocabulary of three terms, t_1, t_2, t_3. Suppose after we ran the above random walk with restart procedure restarting from each one of the three terms we have the following three stationary distributions: $\pi_1 = [0.9, 0.1]$; $\pi_2 = [0.3, 0.5]$; $\pi_3 = [0.09, 0.91]$. If a user submit a query (t_1, t_3) the system will suggest the query q_2 since the vector of scores $\vec{r}$ as defined above results to be $[0.081, 0.091]$, thus q_2 ranks higher in this case.

3. LONG-TAIL RECOMMENDATIONS

We next report few examples from a preliminary assessment of the model. We will report a deeper analysis in an extended version of the paper.

We build our model starting from a 6 months anonymized query logs (from January to June 2010) from Yahoo! US. The resulting TQ-Graph consists of 6,261,105 Term nodes and 28,763,637 Query nodes. We have $83,808,761$ edges from Term nodes to Query nodes and $56,250,874$ edges from Query nodes to Query nodes.

We next report recommendations for queries that are not present in our query log, thus queries for which previously known methods could not provide recommendation. We report the first five results obtained for each query.

Query: Social Katz index

Suggested Query	Score
katz index of activities of daily living	2.7 e^{-09}
modified barthel index	8.6 e^{-13}
barthel index score	6.4 e^{-13}
modified barthel index score	8.8 e^{-14}
youtube	4.3 e^{-19}

The first query shown is *Social Katz index.* The first four suggested queries are highly related with the input query. Both Katz index and Barthel index are indeed indexes used to measure performance in basic activities of daily living. The last query suggested is not related with the query. However, we observe that its score is at least roughly five orders of magnitude smaller than the one of other results.

The second query shown is *Menu restaurant design.* In this case all the reported queries are related to the input query. It is interesting to notice that the third result is *restaurant menu design* that is a more correct way to formulate the same query. Our method is insensitive to permutation of terms in the query.

Query: Menu restaurant design

Suggested Query	Score
free restaurant design software	4.9 e^{-12}
free restaurant kitchen layout and design	4.8 e^{-12}
restaurant menu design	4.7 e^{-12}
restaurant menu design software	4.7 e^{-12}
restaurant menu design samples	4.4 e^{-12}

Obviously the method must perform reasonably well also for head-queries. Next we show the first 5 results obtained by querying for *"Bill Clinton"*. Even if this query is not in the long tail, and we have a query node `Bill Clinton` in V_Q, it is interesting to observe that, by splitting the query in terms, computing the random walks, and then combining the results, we do not loss precision. In fact, we notice that all the top 5 queries are related to the input query. Moreover, we observe that the results are highly diversified: we have clinton's speeches, scandal, foundation and biography.

Query: Bill Clinton

Suggested Query	Score
president bill clinton speeches	8.8 e^{-11}
famous bill clinton quotes	8.4 e^{-11}
monica lewinsky and bill clinton scandal	8.4 e^{-11}
bill clinton foundation website	8.3 e^{-11}
former president bill clinton biography	8.1 e^{-11}

4. EFFICIENCY

It is needless to say that the suggestions must be computed efficiently since a recommender system should work online. Our effort and contribution in this project is also devoted to the efficiency aspect, as it will be described in an extended version of this work.

We can anticipate here that our model is particularly suitable for efficient implementation. In fact, we can store precomputed information enabling to efficiently compute suggestions in a way that recall the process of intersecting posting lists. More in details, we store the stationary distribution of each random walk for each term as it was a posting list by sorting queries' ids by their scores. We also adopt state-of-the-art bucketing, approximations and thresholding techniques in order to considerably reduce the space requirements. Finally the recommendations for a given query are computed by scanning and intersecting the posting lists corresponding to its terms.

Acknowledgement

This research has been partially funded by the EU CIP PSP-BPN ASSETS Project. Grant Agreement no. 250527.

5. REFERENCES

[1] Ricardo Baeza-yates, Carlos Hurtado, and Marcelo Mendoza. Query recommendation using query logs in search engines. In *Proc. of ClustWeb'04*.

[2] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. The query-flow graph: model and applications. In *Proc. CIKM'08*.

[3] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. From 'dango' to 'japanese cakes': Query reformulation models and patterns. In *Proc. WI'09*.

[4] Doug Downey, Susan Dumais, and Eric Horvitz. Heads and tails: studies of web search with common and rare queries. In *Proc. SIGIR'07*.

[5] Fabrizio Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 1(1-2):1–174, 2010.

[6] Yang Song and Li-wei He. Optimal rare query suggestion with implicit user feedback. In *Proc. WWW'10*.

[7] Hanghang Tong and Christos Faloutsos. Center-piece subgraphs: problem definition and fast solutions. In *Proc. KDD'06*.

[8] Ji-Rong Wen, Jian-Yun Nie, Hong-Jiang Zhang, and Hong-Jiang Zhang. Clustering user queries of a search engine. In *Proc. WWW'01*.