Aging Effects on Query Flow Graphs for Query Suggestion

Ranieri Baraglia
ISTI - CNR
Via G. Moruzzi, 1
56124 Pisa, Italy
r.baraglia@isti.cnr.it

Carlos Castillo
Yahoo! Research Barcelona
Avinguda Diagonal 177
08018 Barcelona
chato@yahoo-inc.com

Debora Donato
Yahoo! Research Barcelona
Avinguda Diagonal 177
08018 Barcelona
debora@yahoo-inc.com

Franco Maria Nardini
ISTI - CNR
Via G. Moruzzi, 1
56124 Pisa, Italy
f.nardini@isti.cnr.it

Raffaele Perego
ISTI - CNR
Via G. Moruzzi, 1
56124 Pisa, Italy
r.perego@isti.cnr.it

Fabrizio Silvestri
ISTI - CNR
Via G. Moruzzi, 1
56124 Pisa, Italy
f.silvestri@isti.cnr.it

ABSTRACT

World Wide Web content continuously grows in size and importance. Furthermore, users ask Web search engines to satisfy increasingly disparate information needs. New techniques and tools are constantly developed aimed at assisting users in the interaction with the Web search engine. Query recommender systems suggesting *interesting queries* to users are an example of such tools. Most query recommendation techniques are based on the knowledge of the behaviors of past users of the search engine recorded in query logs.

A recent query-log mining approach for query recommendation is based on *Query Flow Graphs* (QFG). In this paper we propose an evaluation of the effects of time on this query recommendation model. As users interests change over time, the knowledge extracted from query logs may suffer an aging effect as new interesting topics appear. In order to validate experimentally this hypothesis, we build different query flow graphs from the queries belonging to a large query log of a real-world search engine. Each query flow graph is built on distinct query log segments. Then, we generate recommendations on different sets of queries. Results are assessed both by means of human judgments and by using an automatic evaluator showing that the models inexorably age.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - *Data Mining*; H.4.3 [**Information Systems Applications**]: Communications Applications

General Terms

Algorithms

CIKM'09, November 2–6, 2009, Hong Kong, China.

Keywords

Query Flow Graph, Query Suggestion, Topic Drift, Aging Effects, Effectiveness in Query Recommendations

1. INTRODUCTION

In the last years, all web search engines have started to provide users with query suggestions to help them to quickly satisfy their needs. One of the main difficulties users find when they use a web search engine, is correctly formulating their needs in a short text query. Translating human thoughts into a concise set of keywords is in fact not straightforward. Doing the opposite, i.e., translating a few keywords back into a human information need is even more difficult. Query recommendation techniques are based on the knowledge of the behaviors of past users of the search engine.

A successfully query-log mining approach for generating useful query recommendation based on *Query Flow Graphs* (QFGs) [2], was recently proposed in [3]. The QFG model aggregates information in a query log by providing a markov-chain representation of the query reformulation process followed by users trying to satisfy the same information need. This paper aims at extending the QFG model by providing a methodology for dealing efficiently with evolving data. The interests of search engine users change in fact over time. New topics may become bursty popular, while others that focused for some time the attention of crowds can suddenly lose importance. The knowledge extracted from query logs can thus suffer an aging effect, and the models used for recommendation rapidly becoming unable to generate useful and interesting queries.

In order to validate our claims and assess our methodology, we build different query flow graphs from the queries belonging to a large query log of a real-world search engine, and we analyze the quality of the recommendation models devised from these graphs to show that they inescapably age.

The paper is organized as follows. Section 2 discusses related works, while Section 3 introduces the concept of query flow graph. The data used for the experiments are described in Section 4, while their analysis finalized to the evaluation of aging effects on the recommendation models is discussed in Section 5. Finally, Section 6 draws some conclusions and outlines future work.

2. RELATED WORK

Different approaches have been proposed in recent years that use query logs to mine wisdom of the crowds for query suggestion. Bruno et al. in [4] use an association rule mining algorithm to devise query patterns frequently co-occurring in user sessions, and a query relations graph including all the extracted patterns is built. A click-through bipartite graph is then used to identify the concepts (synonym, specialization, generalization, etc.) used to expand the original query. Jones et al. in [6] introduce the notion of query substitution or query rewriting, and propose a solution for sponsored search. Such solution relies on the fact that in about half sessions the user modifies a query with another which is closely related. Such pairs of reformulated queries are mined from the log and used for query suggestion. Baeza-Yates et al. [1] use a k-means algorithm to cluster queries by considering both topics and text from clicked URLs. Then the cluster most similar to user query is identified, and the queries in the cluster with the highest similarity and attractiveness (i.e. how much the answers of the query have attracted the attention of past users), are suggested.

3. THE QUERY FLOW GRAPH

A *Query Flow Graph* (QFG) is a compact but powerful representation of the information contained in a query log. It has been applied successfully to model user interactions with a web search engine and for a number of practical applications as segmenting physical sessions into logical sessions or query recommendation. In this Section we briefly recall some practical steps to infer a QFG from a query log.

Boldi et al. in [2] define a *Query Flow Graph* (QFG) as a directed graph $G = (V, E, w)$ where:

- $V = Q \cup s, t$, is the set of distinct queries Q submitted to the search engine enriched with two special nodes s and t, representing a *starting state* and a *terminal state* which can be seen as the begin and the end of all the chains;
- $E \subseteq V \times V$ is the set of *directed edges*;
- $w : E \rightarrow (0..1]$ is a weighting function that assigns to every pair of queries $(q, q') \in E$ a weight $w(q, q')$.

Each query is represented by a single node independently of its frequency, or of the number of distinct users who issued it. The two special nodes s and t capture respectively the beginning and the end of a chain. The Query Flow Graph is built according to the algorithm presented by Boldi et al. in [2].

4. EXPERIMENTAL FRAMEWORK

Our experiments have been conducted on the AOL query log. The AOL data-set contains about 20 million queries issued by about 650,000 different users, submitted to the AOL search portal over a period of three months from 1st March, 2006 to 31st May, 2006. To assess the aging effects we conducted several experiments to evaluate the impact of different factors. The log has been split into three different *segments*. Two of them have been used for training and the third one for testing. The three segments correspond to the three different months of users activities recorded in the query log. We fixed the test set – i.e. the set of queries from which we generate recommendations – to be the queries submitted in the last month. Table 1 shows the number of nodes and edges of the different graphs corresponding to each query log segment used for training:

time window	id	nodes	edges
March 06	$\mathcal{M}_1$	3,814,748	6,129,629
April 06	$\mathcal{M}_2$	3,832,973	6,266,648

Table 1: Number of nodes and edges for the data-graphs corresponding to the two different training segments.

It is important to remark that we have not re-trained the classification model for the assignment of weights associated with QFG edges. We reuse the one that has been used in [2] for segmenting users sessions into query chains[1]. Once the QFG has been built, the query recommendation methods are based on the probability of being at a certain node after performing a random walk over a query graph. This random walk starts at the node corresponding to the query for which we want to generate a suggestion. At each step, the random walker either remains in the same node with a probability α, or it follows one of the out-links with probability equal to $1 - \alpha$; in the latter case, out-links are followed proportionally to $w(i, j)$. In all the experiments we computed the stable vector of the random walk on each QFG by using $alpha = 0.85$. Actually, the stable vector is computed according to a Random Walk with Restart model [8]. Instead of restarting the random walk from a query chosen uniformly at random, we restart the random walk only from a given set of nodes. This is done by using a preference vector v, much in the spirit of the Topic-based PageRank computation [5], defined as follows. Let $q_1, \ldots, q_n$ be a query chain (q_1 is the most recently submitted query). The preference vector v is defined in the following way: $v_q = 0$ for all $q \notin q_1, \ldots, q_n$ and $v_{q_i} \propto \beta^i$. β is a weighting factor that we set in all of our experiments to be $\beta = 0.90$.

5. EVALUATING THE AGING EFFECT

The main goal of this paper is to show that time has some negative effects on the quality of query suggestions generated by QFG-based models. It is also worth remarking that we can safely extend the discussion that follows also to suggestion models different from QFG-based ones. As a matter of fact, the presence of "*bursty*" [7] topics could require frequent model updates whatever model we are using. To validate our hypothesis about the aging of QFG-based models we have conducted experiments on models built on the two different segments according to the procedure described in the above section.

In order to assess the various reasons why a QFG-based model ages we have considered, for each segment, two classes of queries, namely $\mathcal{F}_1$, and $\mathcal{F}_3$, which respectively correspond to queries having a strong decrease and a strong increase in frequency. $\mathcal{F}_1$ is the set of the 30 queries that are among the 1,000 most frequent queries in the first month ($\mathcal{M}_1$) but whose frequency has had the greater drop in the last month covered by the query log ($\mathcal{M}_3$). Conversely, $\mathcal{F}_3$ is the set of the 30 queries among the 1,000 most frequent queries in the test log $\mathcal{M}_3$ whose frequency has the greater

[1]We thank the authors of [2] for providing us their model.

drop in the first part of the log $\mathcal{M}_1$. Actually, to make the assessment more significant, we do not include queries that are too similar, and we do not include queries containing domain names within the query string. Figure 1 graphically show where the selected queries for each class fall when we plot the popularity of the top-1000 most frequent queries in $\mathcal{M}_3$ by considering query ids assigned according to frequencies in $\mathcal{M}_1$.

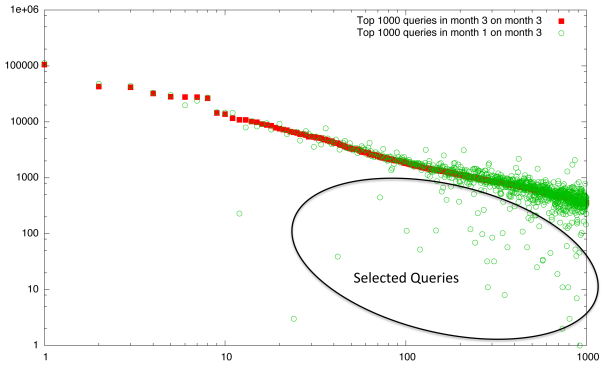

Figure 1: Queries in $\mathcal{F}_3$. The set of top 1,000 queries in $\mathcal{M}_3$ compared with the same set projected on $\mathcal{M}_1$. Query identifiers are assigned according to frequencies in $\mathcal{M}_3$. The circled area in the plot highlights the zone from where $\mathcal{F}_3$ was drawn.

Some examples of queries in $\mathcal{F}_1$ are: "*shakira*", "*americanidol*", "*nfl*". Some other examples of queries in $\mathcal{F}_3$ are: "*mothers day gift*", "*memorial day*", "*da vinci code*". The queries are related to particular events in March 2006, for instance singer Shakira in March 2006 released a music album, and in May 2006 the movie adaptation of the popular book "Da Vinci Code" was published.

We selected two distinct sets because we want to assess the effectiveness of recommendations for both new or emerging query topics in the test log (i.e. queries in $\mathcal{F}_3$), and for queries that are frequent in the first month but poorly represented (or absent) in the test month (i.e. queries in $\mathcal{F}_1$). The first evaluation we perform is a *human-based assessment* of the quality of query suggestions generated by models trained on the two different segments. From each query in $\mathcal{F}_1$ and $\mathcal{F}_3$ we generated the top 20 recommendations using four different sets of QFG-based models: three of them are filtered with different threshold values (0.5, 0.65, and 0.75), one is generated without filtering (threshold 0). Each set consists of QFGs built on either $\mathcal{M}_1$, or $\mathcal{M}_2$. The generated recommendations were manually evaluated and classified as **useful** and **not useful**. We consider useful a recommendation that undoubtedly interprets the possible intent of the user better than the original query.

Table 3 shows the results of the human assessment performed by counting, for each query and the three different threshold levels, the number of useful suggestions. We averaged the counts over all the queries evaluated. For each training period we show the average number of useful suggestion for queries in the three different groups, i.e. $\mathcal{F}_1$, $\mathcal{F}_3$, and $\mathcal{F}_1 \cup \mathcal{F}_3$.

From the table we can draw some interesting conclusions. First, the performance of the models built from $\mathcal{M}_1$ and $\mathcal{M}_2$ are quite similar (column $\mathcal{F}_1 \cup \mathcal{F}_3$). This might seem

filtering threshold	average number of useful suggestions on $\mathcal{M}_1$			average number of useful suggestions on $\mathcal{M}_2$		
	$\mathcal{F}_1$	$\mathcal{F}_3$	$\mathcal{F}_1 \cup \mathcal{F}_3$	$\mathcal{F}_1$	$\mathcal{F}_3$	$\mathcal{F}_1 \cup \mathcal{F}_3$
0	2.51	2.02	2.26	2.12	2.46	**2.29**
0.5	3.11	2.69	**2.9**	2.88	2.87	2.87
0.65	3.02	2.66	**2.84**	2.8	2.71	2.76
0.75	3	2.64	**2.82**	2.72	2.68	2.7

Table 3: Model aging statistics varying the model type and the temporal window. Results were manually assessed. Best results are represented in bold typeface.

a counterexample to the hypothesis that the models age. Actually, by breaking down the overall figure into separate figures for $\mathcal{F}_1$ and $\mathcal{F}_3$ we can observe that for all the queries in $\mathcal{F}_3$ the suggestions built from $\mathcal{M}_2$ are more useful than those built on $\mathcal{M}_1$. Furthermore, by inspecting some of the suggestions generated for the queries shown in Table 2, it is evident that some of the suggestions are "*fresher*" (i.e. more up-to-date) in the case of a model built on $\mathcal{M}_2$ than those obtained on models built on $\mathcal{M}_1$. This is particularly true for queries in $\mathcal{F}_3$.

When we performed the assessment of the suggestions we noted a phenomenon regarding the scores computed on the different QFGs by the random walk-based method. Let us consider again the results shown in Table 2 and let us look at the suggestions, with the relative scores, computed for 6 queries (3 queries from $\mathcal{F}_1$ and 3 queries from $\mathcal{F}_3$) on $\mathcal{M}_1$ and $\mathcal{M}_2$. As we go further down the list sorted by score, when the quality of the suggestions starts to degrade, we often observe that the useless suggestions are associated with the same low score values, e.g. "*regions banking*", "*aol email only*", "*adultactioncamcom*" are three different (and useless) query suggestions for the query "*harley davidson*" whose QFG computed score is always 1394.

From the above observation we make the following hypothesis that we will use to derive an automatic evaluation methodology to assess the "usefulness" of suggestions:

> *when a QFG-based query recommender system gives the same score to consecutive suggestions, these recommendations and the following ones having a lower score are very likely to be useless..*

A QFG-based recommender system recommends queries by computing a random walk with restart on the model. At each step, the random walker either remains in the same node with a probability α, or it follows one of the out-links with probability equal to $1 - \alpha$. Out-links are followed proportionally to $w(i, j)$. Let us suppose the recommender system start recommending more than k queries sharing the same score for the given query q. On the QFG model it means that the query q has more than k out-links sharing the same probability ($w(i, j)$). Due to the lack of information the system is not able to assign a priority to the k recommended queries. This is the reason why we consider these recommendations as "useless". This heuristic considers useful k query recommendations if the suggestions following the top-k recommended queries have equal scores associated with them. Consider again the case of the query "harley davidson", we have six queries with different scores and then the remaining queries (for which the associated scores are equal) are clearly useless.

Query Set	Query	$\mathcal{M}_1$		$\mathcal{M}_2$	
$\mathcal{F}_3$	da vinci	49743	da vinci's self portched black and white	73219	da vinci and math
		47294	the vitruvian man	33769	da vinci biography
		35362	last supper da vinci	31383	da vinci code on portrait
		31307	leonardo da vinci	29565	'ying machines
		30234	post it	28432	inventions by leonardo da vinci
		30234	handshape 20stories	26003	leonardo da vinci paintings
				23343	friends church
				23343	jerry c website
$\mathcal{F}_1$	harley davidson	5097	harley davidson ny	5749	harley davidson premium sound system owners manual
		2652	american harley davidson	3859	automatic motorcycles
		2615	2002 harley davidson ultra classic	3635	harley davidson credit
		2602	adamec harley davidson	3618	cherokee harley davidson
		2341	air ↓ght	2103	harley davidson sporster
		2341	928 zip code	1965	2002 harley davidson classic
		2341	antispy ware	1394	regions banking
				1394	aol email only
				1394	adultactioncamcom

Table 2: Some examples of recommendations generated on different QFG models. Queries used to generate recommendations are taken from different query sets.

We perform the automatic analysis described above to the 400 most frequent queries in the third month for which recommendations were generated on models built on either $\mathcal{M}_1$ or $\mathcal{M}_2$. For all the experiments we set $k = 3$. Table 4 shows that according to this measure of quality filtered models works better than unfiltered ones. The filtering process reduces the "noise" on the data and generates more precise knowledge on which recommendations are computed. Furthermore, the increase is quite independent from the threshold level, i.e. by increasing the threshold from 0.5 to 0.75 the overall quality is, roughly, constant.

filtering threshold	average number of useful suggestions on $\mathcal{M}_1$	average number of useful suggestions on $\mathcal{M}_2$
0	2.84	2.91
0.5	5.85	6.23

Table 4: Recommendation statistics obtained by using the automatic evaluation method on a relatively large set of 400 queries drawn from the most frequent in the third month.

We further break down the overall results shown in Table 4 to show the number of queries on which the QFG-based model generated a given number of useful suggestions. To highlight more the aging effect we show in Figure 2 the total number of queries having at least a certain number of useful recommendation. For example, the third bucket shows how many queries have at least three useful suggestions. For each bucket, results for $\mathcal{M}_2$ are always better than the ones for $\mathcal{M}_1$. Furthermore, for Figure 2 we can observe that a model trained on $\mathcal{M}_2$ has a larger percentage of queries for which the number of useful suggestions is at least 4. This confirms our hypothesis that QFG-based recommendation models age.

6. CONCLUSION AND FUTURE WORK

In this paper we have studied the effect of time on recommendations generated using *Query Flow Graphs* [2] (QFGs). We have shown that the interests of search-engine users change over time and new topics may become popular, while other that focused for some time the attention of the crowds can suddenly loose importance. The knowledge extracted from query logs can thus suffer from an aging effect, and the models used for recommendations becoming unable to generate useful and interesting suggestions.

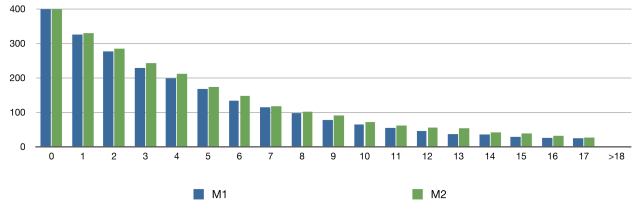

Figure 2: Histogram showing the total number of queries having at least a certain number of useful recommendations. For instance the third bucket shows how many queries have at least three useful suggestions. Results are computed automatically.

7. REFERENCES

[1] R. Baeza-Yates, C. Hurtado, and M. Mendoza. *Query Recommendation Using Query Logs in Search Engines*, volume 3268/2004 of *Lecture Notes in Computer Science*, pages 588–596. Springer Berlin / Heidelberg, November 2004.

[2] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *In Proc. CIKM'08*, pages 609–618, New York, NY, USA, 2008. ACM.

[3] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *In Proc. WSCD'09*, pages 56–63, New York, NY, USA, 2009. ACM.

[4] B. M. Fonseca, P. Golgher, B. Pôssas, B. Ribeiro-Neto, and N. Ziviani. Concept-based interactive query expansion. In *In Proc. CIKM'05*, pages 696–703, New York, NY, USA, 2005. ACM.

[5] T. H. Haveliwala. Topic-sensitive pagerank. In *In Proc. WWW'02*, pages 517–526, New York, NY, USA, 2002. ACM.

[6] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *In Proc. WWW'06*, pages 387–396, New York, NY, USA, 2006. ACM Press.

[7] J. Kleinberg. Bursty and hierarchical structure in streams. In *In Proc. KDD'02*, pages 91–101, New York, NY, USA, 2002. ACM.

[8] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *In Proc. ICDM'06*, pages 613–622, Washington, DC, USA, 2006. IEEE Computer Society.