Simrank++: Query Rewriting through Link Analysis of the Click Graph (Poster)

Ioannis Antonellis
Computer Science Dept.
Stanford University
CA, 94305, USA
antonell@cs.stanford.edu

Hector Garcia-Molina
Computer Science Dept.
Stanford University
CA, 94305, USA
hector@cs.stanford.edu

Chi-Chao Chang
Yahoo! Inc.
Sunnyvale, CA, 94089
chichao@yahoo-inc.com

ABSTRACT

We focus on the problem of query rewriting for sponsored search. We base rewrites on a historical click graph that records the ads that have been clicked on in response to past user queries. Given a query q, we first consider Simrank [2] as a way to identify queries similar to q, i.e., queries whose ads a user may be interested in. We argue that Simrank fails to properly identify query similarities in our application, and we present two enhanced versions of Simrank: one that exploits weights on click graph edges and another that exploits "evidence." We experimentally evaluate our new schemes against Simrank, using actual click graphs and queries form Yahoo!, and using a variety of metrics. Our results show that the enhanced methods can yield more and better query rewrites.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms, Experimentation, Theory

Keywords

sponsored search, link analysis, similarity metric, click graph

1. INTRODUCTION

In sponsored search, paid advertisements (ads) relevant to a user's query are shown above or along-side traditional web search results. The placement of these ads is in general related to a ranking score which is a function of the semantic relevance to the query and the advertiser's bid.

Ideally, a sponsored search system has access to a database of available ads and a set of bids. Conceptually, each bid consists of a query q, an ad α, and a price p. With such a bid, the bidder offers to pay if the ad α is both displayed and clicked when a user issues query q. For many queries, there are not enough direct bids, so the sponsored search system attempts to find other ads that may be of interest to the user who submitted the query. Even though there is no direct bid, if the user clicks on one of these ads, the search engine will make some money (and the advertiser will receive a customer). The challenge is then to find ads related to incoming queries that may yield user click throughs. For a variety of practical and historical reasons, the sponsored search system is often split into two components. A front-end takes an input query q and produces a list of *re-writes*, i.e., of other queries that are "similar" to q. The query and its rewrites are then considered by the back-end, which displays ads that have bids for the query or its rewrites. The split approach reduces the complexity of the back-end, which has to deal with rapidly changing bids. The work of finding relevant ads, indirectly through related queries, is off-loaded to the front-end.

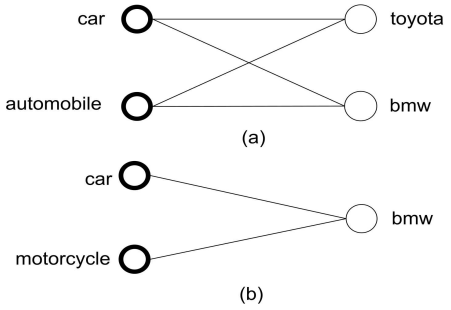

Figure 1: Sample complete bipartite graphs ($K_{2,2}$ and $K_{1,2}$) extracted from a click graph.

At the front-end, queries can be rewritten using a variety of techniques developed for document search. However, these techniques often do not generate enough useful rewrites. Part of the problem is that in our case "documents" (the ads) have little text, and queries are very short, so there is less information to work with, as compared with larger documents. Another problem is that there are relatively few queries in the bid database, so even if we found all the textually related ones, we may not have enough. Thus, it is important to generate additional rewrites, using other techniques.

Our focus is on query rewrites based on the recent history of ads displayed and clicked on. The back-end generates a historical *click graph* that records the clicks that were generated by ads when a user inputs a given query. The click graph is a weighted bi-partite graph, with queries on one side and ads on the other. The schemes we present analyze the connections in the click graph to identify rewrites that may be useful. Our techniques identify not only queries that are directly connected by an ad but also queries that are more

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

Table 1: Query-query similarity scores for the sample click graphs of Figure 1. Scores have been computed by Simrank with $C_1 = C_2 = 0.8$

Iteration	sim("car", "automobile")	sim("car", "motorcycle")
1	0.4	0.8
2	0.56	0.8
3	0.624	0.8
4	0.6496	0.8
5	0.65984	0.8
6	0.663936	0.8
7	0.6655744	0.8

indirectly related. Our techniques are based on the notion of SimRank [2], which can compute query similarity based on the connections in a bi-partite click-graph. However, in our case we need to extend SimRank to take into account the specifics of our sponsored search application.

2. MOTIVATING EXAMPLE

Figure 1 illustrates two sample click graphs. The left nodes are queries issued by users and the right nodes correspond to ads. An edge between a query (left node) and an ad (right node) indicates that at least someone clicked on the ad after issuing that query. If we look at the similarity scores that Simrank computes for the two query pairs car - automobile and car - motorcycle, we can see that sim(car, automobile) is always less than sim(car, motorcycle) no matter how many iterations of the Simrank computation we perform. Table 1 tabulates these scores for the first 7 iterations. In fact, we can prove that sim(car, automobile) becomes eventually equal to sim(car, motorcycle) as we include more iterations.

In this example, the SimRank similarity of car to motorcycle is less than that of car to automobile (if we run limited iterations) or at best they are equal (if we pay the high price of running to convergence). Either result is counterintuitive since there are more ads that connect car and automobile. To give car and automobile higher similarity, we introduce the notion of "evidence of similarity."

3. REVISING SIMRANK

3.1 Evidence-based Simrank

Given two nodes a, b of a bipartite graph, we will denote as evidence(a, b) the evidence existing in G that the nodes a, b are similar. The intuition behind choosing such a function is as follows. We want the evidence score evidence(a,b) to be an increasing function of the common neighbors between a and b. In addition, we want the evidence scores to get closer to one as the common neighbors increase. By multiplying the Simrank scores with the evidence function we fix the anomalies observed before. Although not shown here, in our example, after 2 iterations the evidence-based Simrank score sim(car, automobile) turns out to be 0.42 while sim(car, motorcycle) is 0.4 and thus someone can correctly determine that car is more similar to automobile than to motorcycle.

3.2 Weighted Simrank

We also modify the underlying random walk model of Simrank so that it exploits the edge weights of the click graph. Again we use the evidence scores, but now we perform a different random walk that utilizes the edge weights in its transition probabilities. More details can be found on the extended version of the paper [1].

4. EXPERIMENTS

We conducted experiments to compare the performance of Simrank, evidence-based Simrank and weighted Simrank as techniques for query rewriting. Our baseline was a query rewriting technique based on the Pearson correlation. To evaluate the quality of rewrites, we consider two methods.

The first is a manual evaluation, carried out by professional members of Yahoo!'s editorial evaluation team. Each query – rewrite pair is considered by an evaluator, and is given a score on a scale from 1 to 4, based on their relevance judgment. The query rewrites that were more relevant with the original query assigned a score of 1, and the least related assigned a score of 4. The judgment scores are solely based on the evaluator's knowledge, and not on the contents of the click graph. The evaluation metrics we used were precision/recall (based on the editorial scores), the query coverage (number of queries for which each method manages to provide at least one rewrite) as well as the query rewriting depth (number of query rewrites that a method provides for a given query). In summary, evidence-based simple Simrank outperforms simple Simrank and Pearson both in query coverage, rewriting depth and precision/recall. Weighted Simrank maintains the query coverage and rewriting depth of evidence-based Simrank but substantially boosts the precision of the rewrites.

Our second evaluation method addresses the question of whether our methods made the "right" decision based on the evidence found in the click graph. The basic idea is to remove certain edges from the click graph and to see if using the remaining data our schemes can still make useful inferences related to the missing data (desirability prediction). Weighted Simrank outperforms all other alternatives here.

More details on the evaluation method and the results can be found on the extended version of the paper [1].

5. CONCLUSIONS

We have discussed the problem of query rewriting for sponsored search. We propose Simrank to exploit the click graph structure and we introduce two extensions: one that takes into account the weights of the edges in the click graph, and another that takes into account the "evidence" supporting the similarity between queries. Our experimental results show that weighted-based Simrank is the overall best method for generating rewrites based on a click graph.

Even though our new schemes were developed and tested for query rewriting based on a click graph, we suspect that the weighted and evidence-based Simrank methods could be of use in other applications that exploit bi-partite graphs. We plan to experiment with these schemes in other domains, including collaborative filtering.

6. REFERENCES

[1] I. Antonellis, H. Garcia-Molina, and C. Chang. Simrank++: Query rewriting through link analysis of the click graph. In *Technical Report, url: http://dbpubs.stanford.edu/pub/2007-32*, 2007.

[2] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proc. KDD '02.*