

Automatic Query Recommendation using Click-Through Data

Georges Dupret¹ and Marcelo Mendoza²

¹ Yahoo! Research Latin America
Blanco Encalada 2120, Santiago, Chile
gdupret@yahoo-inc.com

² Department of Computer Science, Universidad de Valparaíso,
Gran Bretaña 1091, Playa Ancha, Valparaíso, Chile
marcelo.mendoza@uv.cl

Abstract. We present a method to help a user redefine a query suggesting a list of similar queries. The method proposed is based on click-through data where sets of similar queries could be identified. Scientific literature shows that similar queries are useful for the identification of different information needs behind a query. Unlike most previous work, in this paper we are focused on the discovery of better queries rather than related queries. We will show with experiments over real data that the identification of better queries is useful for query disambiguation and query specialization.

1 Introduction

Besides the practical difficulties inherent to the manipulation of a large number of pages, the difficulty of ranking documents based on generally very short queries and the fact that query needs are often imprecisely formulated, users are also constrained by the query interface, and often have difficulty articulating a query that could lead to satisfactory results. It happens also that they do not exactly know what they are searching for and select query terms in a trial and error process.

Better formulation of vague queries are likely to appear in the query logs because experienced users tend to avoid vague formulation and issue queries specific enough to discard irrelevant documents. Different studies sustain this claim: [7] report that an independent survey of 40,000 web users found that after a failed search, 76% of users try rephrasing their query on the same search engine. Besides, the queries can be identical or articulated differently with different terms but they represent the same information needs as found and reported in [6, 9].

We propose in this work a simple and novel query recommendation algorithm. While most methods recommend queries related to the original query, this method aims at improving it.

1.1 Related Work

A well known query reformulation strategy is *relevance feedback* [1]. The main idea is to present to a user a list of documents related to an initial query. After examining them, the user selects those which are relevant. Using important terms attached to the selected documents, it is possible to reformulate the original query adding terms to it and *reweighting* the query representation in the document vector model. Recent work also incorporate logs as a useful source data for this strategy [2]. For example, in [5], large query logs are used to construct a surrogate for each document consisting of the queries that were a close match to that document. It is found that the queries that match a document are a fair description of its content. They also investigate whether query associations can play a role in query expansion. In this sense, in [8], a somewhat similar approach of summarizing document with queries is proposed: Query association is based on the notion that a query that is highly similar to a document is a good descriptor of that document. One of the main differences of *relevance feedback* strategies with our approach is that relevance feedback is focused on the improvement of the quality of the documents retrieved. We propose a more flexible strategy, focused on the identification of alternate queries allowing the specialization or generalization of the original query. Finally, we are interested in the identification of better queries more than related queries.

The literature also shows related work based on clustering techniques. For example, Wen et al. [10] suggest a method for recommend queries using clustering techniques over data generated from query logs. Given a particular query, it utilizes the previous session logs of the same query to return pages that most users visit. To get a statistically significant result, it is only applied on a small set of popular queries. Because the engine is a commercial one, more details on the method are uneasy to obtain. On the other hand, Baeza-Yates et al. [4] recommend queries building a term-weight vector for each query. Each term is weighted according to the number of document occurrences and the number of selection of the documents in which the term appears. Then they use clustering techniques to identify related queries. Unfortunately, the method is limited to queries that appears in the log and is biased by the search engine.

Finally, some interesting applications such as *Electronic Roads* [3] use concepts related to query specialization in order to offer to the user better navigation choices. Considering user profile information and meta-data, the application creates dynamic links in each new navigation step to new information units. One of the main differences between *Electronic Roads* and our approach is that the creation of the meta-data is performed by experts, so the proposed method is semi-automatic. Also the system needs a user profile to improve their recommendations. Is our aim to propose an automatic query recommendation method based only in the information registered in the logs, without the user cooperation.

1.2 Contribution

The simple method we propose here aims at discovering alternate queries that improve the search engine ranking of documents: We order the documents selected during past sessions of a query according to the ranking of other past queries. If the resulting ranking is better than the original one, we recommend the associated query.

1.3 Outline

The remainder of this paper is organized as follows. In Sect. 2 we present our query recommendation method. We apply this method in Sect. 3 on the query logs of TodoCL, an important Chilean search engine. Finally we conclude in Sect. 4.

2 Query Recommendation Framework

2.1 Preliminaries

First, we need to introduce some definitions. We understand by **keyword** any unbroken string that describes the query or document content. A **query** is a set of one or more keywords that represent an information need formulated to the search engine. The same query may be submitted several times. Each submission induces a different **query instance**. A **query session** consists of one query and the URLs the user clicked on, while a **click** is a Web page selection belonging to a query session. We also define a notion of consistency between a query and a document:

Definition 1 (Consistency). *A document is **consistent** with a query if it has been selected a significant number of times during the sessions of the query.*

Consistency ensures that a query and a document bear a natural relation in the opinion of users and discards documents that have been selected by mistake once or a few time. Similarly, we say that a query and a set of documents are consistent if each document in the set is consistent with the query.

2.2 Query Recommendation method

Many identical queries can represent different user information needs. Depending on the topic the user has in mind, he will tend to select a particular subgroup of documents. Consequently, the set of selections in a session reflects a sub-topic of the original query. We might attempt to assess the existing correlations between the documents selected during sessions of a same query, create clusters and identify queries relevant to each cluster, but we prefer a simpler, more direct method where clustering is done at the level of query sessions.

Let $D(s_q)$ be the set of documents selected during a session s_q of a query q . If we make the assumption that $D(s_q)$ represents the information need behind q , we might wonder if other queries are consistent with $D(s_q)$ and better rank the documents of $D(s_q)$. If these queries exist, they are potential query recommendations. We then repeat the procedure for each session of the original query, select the potentially recommendable queries that appear in a significant number of sessions and propose them as recommendations to the user interested in q .

We need to introduce a criteria in order to compare the ranking of a set of documents for two different queries. Firstly, we define the **rank** of a document in a query as follows:

Definition 2 (Rank of a Documents). *The rank of document u in query q , denoted $r(u, q)$, is the position of document u in the answer list returned by the search engine.*

We extend this definition to sets of documents:

Definition 3 (Rank of a Set of Documents). *The rank of a set U of documents in a query q is defined as:*

$$r(U, q) = \max_{u \in U} r(u, q) .$$

In other words, the document with the worst ranking determines the rank of the set. Intuitively, if a set of documents achieves a better rank in a query q_a than in a query q_b , then we say that q_a ranks the documents better than q_b . We formalize this as follows:

Definition 4 (Ranking Comparison). *A query q_a ranks better a set U of documents than a query q_b if:*

$$r(U, q_a) < r(U, q_b) .$$

This criteria is illustrated in Fig. 1 for a session containing two documents.

Now, it is possible to recommend queries comparing their rank sets. We can formalize the method as follows:

Definition 5 (Recommendation). *A query q_a is a recommendation for a query q_b if a significant number of sessions of q_a are consistent with q_b and are ranked better by q_a than by q_b .*

The recommendation algorithm induces a directed graph between queries. The original query is the root of a tree with the recommendations as leaves. Each branch of the tree represents a different specialization or sub-topic of the original query. The depth between a root and its leaves is always one, because we require the recommendations to improve the associated document set ranking.

Finally, we observe that nothing prevents two queries from recommending each other:

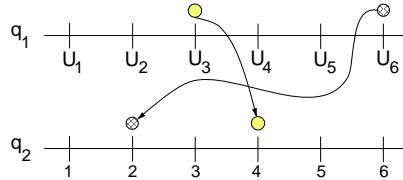


Fig. 1. Comparison of the ranking of two queries. A session of the original query q_1 contains selections of documents U_3 and U_6 appearing at position 3 and 6 respectively. The rank of this set of document is 6 by virtue of Def. 3. By contrast, query q_2 achieves rank 4 for the same set of documents and therefore qualifies as a candidate recommendation.

Definition 6 (Quasi-synonyms). *Two queries are quasi-synonyms when they recommend each other.*

We will see in the following section that this definition leads to queries that are indeed semantically very close.

3 Experimental Results

In this section we present the evaluation of the method introducing a brief descriptive analysis of the results and showing a user evaluation experiment. Firstly we will describe the data used for the experiments.

3.1 Data

The algorithm implied by Def. 5 was easy to implement using log data organized in a relational database. We used the logs of the `TodoCL` search engine (www.todocl.cl) for a period of three months. `TodoCL` is a search engine that mainly covers the `.cl` domain (Chilean web pages) and some pages included in the `.net` domain that are related with Chilean ISP providers. It indexed over 3,100,000 Web pages and has currently over 50,000 requests per day.

Over three months the logs gathered 20,563,567 requests, most of them with no selections: Meta search engines issue queries and re-use the answer of `TodoCL` but do not return information on user selections. A total of 213,540 distinct queries lead to 892,425 registered selections corresponding to 387,427 different URLs. There are 390,932 query sessions. Thus, in average users clicked 2,28 pages per query session and 4,17 pages per query.

3.2 Descriptive Analysis of the Results

We intend to illustrate with two examples that the recommendation method has the ability to identify sub-topics and suggest query refinement. Fig. 2 shows

the recommendation graphs for the queries **Valparaiso** and **Fiat**. The query **Valparaiso** requires some contextual explanation. Valparaiso is an important harbor city, with various universities. It is also the home for the *Mercurio*, the most important Chilean newspaper. It also recommends some queries that are typical of any city of some importance like city hall, municipality and so on. The more potentially beneficial recommendations have a higher link number. For example $9/33 \simeq 27\%$ of the users would have had access to a better ranking of the documents they selected if they had searched for **university** instead of **Valparaiso**. This also implicitly suggests to the user the query **university Valparaiso**, although we are maybe presuming the user intentions. A more complex recommendation graph is associated to the query **Fiat**. The user who issued the query **Fiat** is recommended essentially to specify the car model he is interested in, if he wants spare parts, or if he is interesting in selling or buying a fiat. Note that such a graph also suggests to a user interested in – say – the history or the profitability of the company to issue a query more specific to his needs.

We already observed that two queries can recommend each other. We show in Table 1 a list of such query pairs found in the logs. We reported also the number of original query sessions and number of sessions enhanced by the recommendation so as to have an appreciation of the statistical significance of the links. We excluded the mutual recommendation pairs with less than 2 links. For example, in the first row, out of the 13 sessions for **ads**, 3 would have been better satisfied by **advert**, while 10 of the 20 sessions for **advert** would have been better satisfied by **ads**.

→	query	query	←
3/13	ads	advert	10/20
3/105	cars	used cars	2/241
34/284	chat	sports	2/13
2/21	classified ads	advertisement	2/20
4/12	code of penal proceedings	code of penal procedure	2/9
3/10	courses of english	english courses	2/5
2/27	dvd	musical dvd	2/5
2/5	family name	genealogy	2/16
3/9	hotels in santiago	hotels santiago	2/11
5/67	mail in Chile	mail company of Chile	2/3
7/15	penal code	code of penal procedure	2/9
8/43	rent houses	houses to rent	2/14
2/58	van	light truck	3/25

Table 1. Examples of “Quasi-synonym” queries recommend each other.

We can see that the proposed method generates a clustering *a posteriori* where different sessions consisting of sometimes completely different sets of

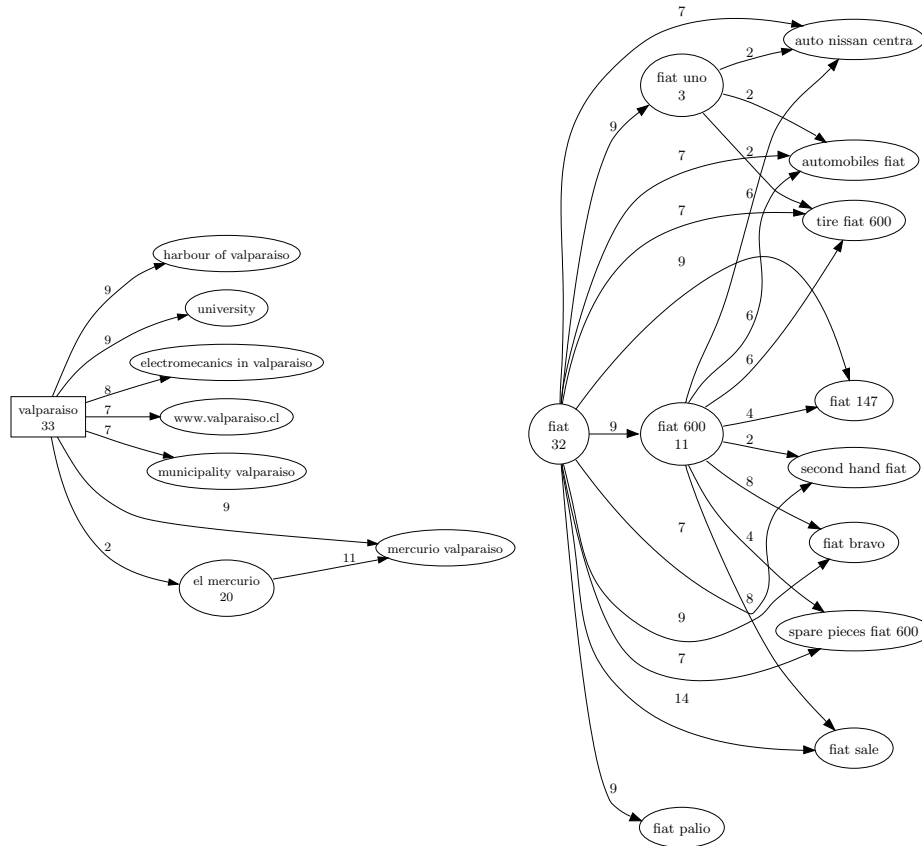


Fig. 2. Queries **Valparaiso**, **Fiat** and associated recommendations. The node number indicate the number of query session. The edge numbers count the sessions improved by the pointed query.

documents end up recommending a same query. This query can then label this group of session.

3.3 User Evaluation

We will analyze user evaluations of the quality of query recommendations. We presented to a group of 19 persons of different backgrounds ten recommendation trees similar to Fig. 2, selected randomly from all the trees we extracted from the logs. Obviously we discarded queries with a small number of sessions and queries with a too large number of recommendations. We asked two questions to the participants:

1. What percentage of recommendations are relevant to the original query?

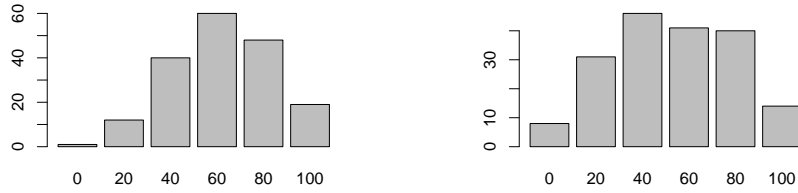


Fig. 3. The figure on the left is related to the first question raised to the participants. It reports on abscissa the percentage of relevant recommendations and in ordinate the number of participant votes. On the right figure, we plotted the results for improved recommendations corresponding to the second question.

2. According to our intuition, what is the percentage of recommendations that will improve a typical user query?

In figure 3 we show the distribution of the opinions for the both questions. We used two factors analysis of variance with no interaction to test whether there is a large variation between participants opinions and between queries. For the first question concerning the relevance of the recommendations, the p -values of the variation induced by the participants is 0.5671 and by the queries is 0.1504, leading to accepting the hypothesis H_0 that none of these variations is statistically significant. The same conclusion holds for question 2 about whether the recommendations might improve the original query. The p -values in this case are 0.9991 and 0.2130. This shows that no participant was over or under estimating the relevance and the improvement percentages systematically, and that no query is particularly worse or better than the other.

The recommendations along with the mean of participants answer can be found in Table 2. Some queries and recommendations in this table are specific to Chile: “El Mercurio” is an important national newspaper, “Colmena” is a private health insurance company but the term “colmena” in Spanish also means beehive. It seems that people searching for honey producers were fooled by a link to the “Colmena” health insurance company. Some sessions of the query for “health insurance company” contained links to “Colmena” that appear high in the ranking for “honey bees”. This problem should disappear if we fix a higher consistency threshold 1, which would be possible with larger logs. Folklore and Biology are important activities in the “University of Chile” that users might look for. “Wei” is a computer store.

Query	Relevance	Improvement	Recommended queries
people finders	52%	45%	Chilean finder Argentinian finder OLE search Finder of Chilean people
used trucks	70%	53%	cars offers used cars sales used trucks rentals trucks spare parts
naval battle of Iquique	78%	76%	Arturo Prat biography J. Prieto government treaty of Ancon Chilean navy
computers	74%	61%	<i>sky Chile</i> Wei motherboards notebook
dictionary	41%	34%	English dictionary dictionary for technology look up words in a dictionary tutorials
El Mercurio	62%	52%	El Mercurio of December <i>currency converter</i> El Mercurio de Valparaiso El Mercurio de Antofagasta
health insurance companies	58%	55%	Banmedica Colmena <i>honey bees</i> contralory of health services
map of Santiago	81%	68%	map of Chile Santiago city city map of Santiago street map of Santiago
Universidad de Chile	41%	25%	Universidad Catolica university folklore biology
wedding dresses	61%	55%	dress rentals wedding dress rentals party dresses bachelor party

Table 2. Queries used for the experiments and recommendations with strong levels of consistency. Average relevance value is shown in column 2. Average improvement value is shown in column 3. Wrong recommendations are in cursive fonts. Trademarks are in bold fonts.

4 Conclusion

We have proposed a method for query recommendation based on user logs that is simple to implement and has low computational cost. The recommendation method we propose are made only if users are expected to improve. Moreover, the algorithm does not rely on the particular terms appearing in the query and documents, making it robust to alternative formulations of an identical information need. Our experiments show that query graphs induced by our method identify information needs and relate queries without common terms.

Acknowledgements

Marcelo Mendoza was supported by CONICYT Chile, project FONDECYT 1061201.

References

1. J.J. Rochio (1971) Relevance feedback in information retrieval. The SMART Retrieval System - Experiments in Automatic Document Processing, Prentice Hall Inc.
2. Ian Ruthven and Mounia Lalmas and C. J. van Rijsbergen (2003) Incorporating user search behavior into relevance feedback. *JASIST* 54(6):529-549.
3. Georgios Fakas and Antonis C. Kakas and Christos Schizas (2004) Electronic Roads: Intelligent Navigation Through Multi-Contextual Information. *Knowledge Information Systems* 6(1):103-124, Springer.
4. Baeza-Yates, R. and Hurtado, C. and Mendoza, M. (2004) Query Recommendation Using Query Logs in Search Engines. *Current Trends in Database Technology - EDBT 2004 Workshops, LNCS 3268:588-596*, Heraklion, Greece.
5. Bodo Billerbeck and Falk Scholer and Hugh E. Williams and Justin Zobel (2003). Query expansion using associated queries. *CIKM 03, 2-9*, ACM Press, New Orleans, LA, USA.
6. Jansen, M. and Spink, A. and Bateman, J. and Saracevic, T. (1998). Real life information retrieval: a study of user queries on the web. *ACM SIGIR Forum*, 32(1):5-17.
7. NPD (2000). Search and Portal Site Survey, Published by NPD New Media Services.
8. Falk Scholer and Hugh E. Williams (2002). Query association for effective retrieval. *CIKM 02, 324-331*, ACM Press, McLean, Virginia, USA.
9. Silverstein, C. and Henzinger, M. and Hannes, M. and Moricz, M. (1999). Analysis of a Very Large Alta Vista Query Log. *SIGIR Forum* 33(3):6-12, ACM Press.
10. Wen, J. and Nie, J. and Zhang, H. (2001). Clustering User Queries of a Search Engine. *Proc. of the 10th WWW Conference*, Hong Kong.