Journal of Web Engineering, Vol. 0, No. 0 (2003) 000–000

DISCOVERING SEARCH ENGINE RELATED QUERIES USING ASSOCIATION RULES

BRUNO M. FONSECA[1 2] PAULO B. GOLGHER[2]
EDLENO S. DE MOURA[3] BRUNO PÔSSAS[1 2] NIVIO ZIVIANI[1]

[1] *Computer Science Department*
Federal University of Minas Gerais
Belo Horizonte, Brazil
{maciel,bavep,nivio}@dcc.ufmg.br

[2] *Akwan Information Technologies*
Av. Abraão Caram, 430, 4⁰ andar - Pampulha
Belo Horizonte, Brazil
{bruno,golgher,bavep}@akwan.com.br

[3] *Computer Science Department*
Federal University of Amazonas
Manaus, Brazil
edleno@dcc.fua.br

Received (received date)
Revised (revised date)

This work presents a method for online generation of query related suggestions for a Web search engine. The method uses association rules to extract related queries from the log of submitted queries to the search engine. Experimental results were performed on a real log containing more than 2.3 million queries submitted to a commercial search engine. For the top 5 related terms our method presented correct suggestions in 90.5% of the time. Using queries randomly selected from a log we obtained 93.45% of correct suggestions. A study of the user behavior showed that in 92.23% of the clicks on suggestions, users found useful information. The same approach can be used to provide terms to the classic problem of query expansion. For instance, the average precision of the answers of the Google search engine was improved by 23.16% using our aproach as a query expansion method.

Keywords: related queries, association rules, search engines, query expansion

Communicated by: to be filled by the Editorial

1 Introduction

The Web has become an essential source of up-to-date information that covers almost all the topics of human knowledge. However, the task of finding relevant information related to a given topic on the Web is difficult. The information available on the Web is not structured and the useful material related to any desired topic is always mixed with billions of Web pages with little or no interest. In this scenario, Web search engines became one of the most

popular services available on the Web.

Despite recent advances in the technology of search engines there are still many situations where users are contemplated with non-relevant answers. One of the great challenges faced by search engines is the difficulty in uncovering an exact description of the users need, as they usually submit very short and imprecise queries [14, 22].

A popular solution to help users in the task of specifying their information needs is to use relevance feedback techniques [19, 4, 11]. These techniques improve the interactivity of the system by allowing users to inform about the relevance of the answers given to their initial query and the feedback information is used to refine the initial query.

A form of relevance feedback that has recently become popular in search engines is to show a list of *related queries* to the user initial query. For instance, if an user searches for *Madonna* in All the Web search engine[a] the following related queries are presented: *madonna lyrics*, *madonna music*, *madonna mp3* and *madonna wedding*. The presentation of a list of related queries is an interesting feedback alternative because users can explicitly reformulate the query by removing possible ambiguities, or turning the query more specific, or just redirecting the query to another topic related to the initial query. It is also known that users prefer explicit reformulation based on suggestions instead of an automatic reformulations of queries [20]. Despite of its growing popularity, there is a lack of related work in the literature on how to get query related suggestions.

In this paper we present a method for online generation of related queries. The method uses an algorithm for mining association rules from the log of past submitted queries to a search engine. We have mapped the problem of mining association rules in customer transactions introduced by Agrawal, Imielinski and Swami [1] to the problem of finding related queries in a Web search engine. In our experiments we used a log containing more than 2.3 million queries submitted to *TodoBR*[b], a popular Brazilian Web search engine. Using the most popular queries and a set of randomly selected ones, experiments show that more than 90% of the suggestions generated by our method are correct. We carried out experiments to evaluate the users behavior when they are presented with query suggestions, and show that there is a high probability that users who click on suggestions are successful.

The same approach can be used to provide terms to the classic problem of query expansion. Query expansion is considered a way to resolve the problems caused by short and imprecise queries. Query expansion with terms provided by our suggestion method improved the average precision of queries submitted to the Google search engine[c] by 23.16%.

This paper is organized as follows. Section 2 presents the related work. Section 3 introduces our method for mining association rules from query logs and identifying related queries. Section 4 presents experimental results and a study of user behavior on a real search engine that uses our query suggestion method. Section 5 presents conclusions and future work.

2 Related Work

Terms suggestions methods are classified into two classes: document based and log based [12]. Document based suggestions extract the correlation between queries previously submitted by

[a] http://www.alltheweb.com
[b] http://www.todobr.com.br
[c] http://www.google.com

users and the content of documents. Log based suggestions use only the correlation between queries previously submitted to a search engine.

Glance [10] proposes a document based method that generates suggestions by analyzing the relationship between the terms in the documents. The method retrieves the top 10 answers given by a search engine for each query and uses this information to study the relationship among query terms and document terms. This relation is mapped in a graph that may be navigated by users to refine their queries.

Cui, Wen, Nie and Ma [8] suggest a method for finding relations between queries and phrases of documents. They use the hypothesis that the click through information available on search engine logs represents an evidence of relation between queries and documents chosen to be visited by users. This evidence is called cross-reference of documents. Based on this evidence, the authors establish relationships between queries and phrases that occur in the documents chosen. These relationships are then used to expand the initial query or to give query suggestions. This approach can also be used to cluster queries extracted from log files [23]. Cross-reference of documents are combined with similarity functions based on query content, edit distance and document hierarchy to find better clusters. These clusters are used in question answering systems to find similar queries.

The works presented above are based on the idea that there is a relationship between the queries and the textual content of documents selected for these queries. This assumption, which is the main assumption of document based term suggestions, is not always true when we are dealing with the Web. In the Web many documents may contain noisy or non textual information and search engines are using alternative sources of information to rank documents [6](such as link analysis) when the document content may not be a good representation of its topic. In both cases, the information used to identify related queries depends on the search engine results, which is costly and makes the methods highly dependent on the quality of the search engine used in the experiments. An example of this problem is presented in [10], where experiments show that their method performs differently when applied to distinct search engines.

Log based methods avoid this dependence because the information on the relationship between query terms that is extracted exclusively from query logs. These methods do not depend on information from the search engine or the documents content. Huang, Chien an Oyang [12] present a method that is similar to ours by relying exclusively on search engine log files to uncover query relationships. They show that log based methods generate better results than document based methods.

Our method innovates by using association rules to extract related queries from search engines log files. Association rules are widely used to develop high quality recommendation systems in e-commerce applications available in the Web [9, 15]. These applications take user sessions stored at system logs to obtain information about the user behavior to recommend services and products. In our work the same idea is applied to find related queries and provide suggestions to Web search engine users. We find previous search patterns that match the current query and use this information to suggest related queries that may be useful to users.

Another way to guide users in the task of finding relevant information is to develop query expansion techniques [4, 13, 16, 5, 25]. Some query expansion methods can be adapted to give

suggestions of new queries. This strategy is different from finding related queries because the expansion methods construct artificial queries, while in our case we give real related queries formulated by other users that had the same information need in the past. On the other hand, related queries can also be used like a query expansion method. In [8] the authors suggest a method that uses the relationships extracted from search engine log files for query expansion. We will present some experiments showing that our method is also useful to be used as a query expansion.

3 Identifying Related Queries

Our method for identifying related queries is divided in three phases, as shown in Figure 1. In phase 1, search engine logs are analyzed and user sessions are extracted. In the second phase, association rules are mined from the set of user sessions and related queries are identified. In the third phase, unwanted suggestions are cleaned from the list of related queries. We now describe each phase in detail.

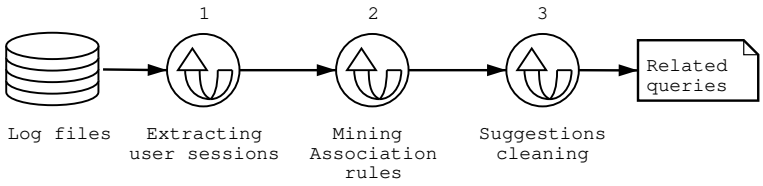

Fig. 1. Identifying related queries

3.1 Extracting user sessions

We call an user session s the set of all queries made by an user within a pre-defined time interval t [d]. The set of user sessions can be extracted directly from search engine Web server logs. The server used in our experiments is the *Squid Proxy*. Figure 2 presents a typical log format as generated by the *Squid Proxy* and Figure 3 presents a real example. Although this format is specific to *Squid*, all web servers generate similar log files.

time elapsed remote-host code/status bytes method URL rfc931 peerstatus/peerhost type

Fig. 2. Squid log format.

The definition of an user session is based on the log information. Each user is identified by the *remote_host* field (its IP address) and the session is defined by the set of queries extracted from the *URL* field. The set of queries is divided into t minutes intervals according to the *time* field. The use of the *remote_host* field may lead to some incorrect sessions, mainly due to users that share the same IP address (e.g., users behind a proxy, dynamic IPs, etc). Some of these incorrect sessions are easily identified by discarting long sessions. In fact, in our experiments, we used sessions with a maximum of 10 queries and longer sessions were discarted. Further, our approach for mining related queries is robust enough to cope with

[d] In our experiments, we used $t = 10$ minutes.

```
1073095205.319 2212 218.70.88.132 TCP_MISS/200 16690 GET http://www.todobr.com.br/cgi-bin/querybas.cgi?query=matrix - DIRECT/200.188.188.214 text/html

1073095272.298 731 200.155.127.37 TCP_MISS/200 17069 GET http://www.todobr.com.br/cgi-bin/querybas.cgi?query=beatles - DIRECT/200.188.188.214 text/html

1073095380.629 2662 200.151.124.194 TCP_MISS/200 17859 GET http://www.todobr.com.br/cgi-bin/querybas.cgi?query=mp3 - DIRECT/200.188.188.215 text/html

1073095276.930 2217 200.222.242.251 TCP_MISS/200 15232 GET http://www.todobr.com.br/cgi-bin/querybas.cgi?query=oasis - DIRECT/200.188.188.213 text/html
```

Fig. 3. Example of entries in the query log.

some incorrect sessions while still maintaining a very high rate of acurracy, as shown by our experiments. The proposal of a precise user session extraction method is beyond the scope of this work. Our aim is to use a method that is simple and runs fast. For more sophisticate approaches we refer the reader to [7].

Once the set of user sessions s is characterized, we start the phase 2 of our method, as described in the next section.

3.2 Mining association rules

The problem of mining association rules in categorical data presented in customer transactions was introduced by Agrawal, Imielinski and Swami [1]. This seminal work gave birth to several investigation efforts [3, 2, 17, 21, 26, 18], resulting in descriptions of how to extend the original concepts and how to increase the performance of the related algorithms.

The original problem of mining association rules was formulated as how to find rules of the form $set_1 \rightarrow set_2$. This rule is supposed to denote affinity or correlation among the two sets containing nominal or ordinal data items. More specifically, such an association rule should translate the following meaning: customers that buy the products in set_1 also buy the products in set_2. Statistical basis is represented in the form of minimum support and confidence measures of these rules with respect to the set of customer transactions.

Before introducing our method to discover related queries we introduce the definition of the mining association rules problem.

3.2.1 Mining association rules problem

Let $I = \{\mathrm{i}_1, \mathrm{i}_2, ..., \mathrm{i}_m\}$ be the set of unique literals (called items) and D a database of transactions. Each transaction $T \in D$ is a non-empty set of items, i.e., $T \neq \emptyset$ and $T \subseteq I$. There is a total ordering among the transaction items, which is based on its lexicographical order, so that $\mathrm{i}_j < \mathrm{i}_{j+1}$, for $1 \leq j \leq t-1$. An *n-itemset* I_j is an ordered set of n unique items, such that $I_j \subseteq I$. Notice that the order among items in I_j follows the aforementioned total ordering. The support count of an itemset I_j is defined as the number of transactions in D that contains I_j.

A itemset I_j is a *frequent itemset* if its support $\sigma(I_j)$ is greater than or equal to a given threshold, which is known as minimum support. As presented in the original Apriori algorithm [3], an n-itemset is frequent if and only if all of its $(n-1)$-itemsets are also frequent.

An association rule is an expression $A \rightarrow B$, where A and B are itemsets. The support of the rule is given as $\sigma(A \cup B)$, and the confidence as $\frac{\sigma(A \cup B)}{\sigma(A)}$ (i.e., the conditional probability that a transaction contains B, given that it contains A).

For a given pair of minimum confidence and support thresholds and a set of transactions, the problem of mining association rules is to find out all the rules that have confidence and support greater than the corresponding thresholds. For the sake of this presentation, the solution of the association rule generation problem is divided into two steps: the first step consists of finding all the frequent itemsets and the second step consists of generating the association rules derived from the frequent sets found in the first step.

3.2.2 Mining related queries

The intuition behind our approach is as follows. During a session, the user defines his information needs submitting a set of queries. If distinct queries cooccur in many sessions for distinct users, this may be an indication that these queries are related.

The mining related queries problem can be mapped to the mining association rules problem in a straightforward way. We map a query into an item and an user session into a transaction. Our approach is based on an efficient algorithm for association rule mining called CHARM [27], which has been adapted to handle queries and sessions instead of items and transactions, respectively. We only determine association rules for the 2-itemsets, i.e., we are interested in correlations between pairs of queries. This simple definition allows our approach to compute the relation between queries in an extremely fast way, which means that new association rules can be updated periodically to identify new groups of related queries. This feature is important since the topics searched on the Web are dynamic and new relations may arise every day.

Once the association rules were determined, we build an inverted file [4, 24] for the discovered rules. An inverted file is typically composed of a *vocabulary* and a set of *inverted lists*. The vocabulary contains all distinct queries found in the query logs. For each query I_k in the vocabulary, there is an inverted list containing all queries I_l related to I_k, and its respective confidence values $c_{k,l}$. Thus, inverted lists consists of pairs of $< I_l, c_{k,l} >$ values, sorted by confidence. This means that the queries for which the confidence values are the greatest can be found quickly.

3.3 Cleaning unwanted suggestions

The main idea of a suggestion system is to remove ambiguities, turn queries more specific or redirect users to another topic that is related to the initial query. The output of the previous phase may result in some related queries that do not fulfill any of these desired characteristics. Thus, this third phase of our method identify this kind of suggestion (we call then unwanted suggestions) using two simple heuristics:

- Suggestions that are plural forms of the original query. For instance, the suggestion "games" for the query "game".

- Suggestions that are substrings of the original query. For instance, the suggestion "games" for the query "free games".

To increase the precision of the suggestions and remove some ambiguities, we also remove stop words[e] during all phases of our method.

4 Experimental Results

In this section we present the experiments carried out to assess the effectiveness of our query suggestion method. First, we manually inspected the quality of the generated suggestions for the top 95 most popular queries of the TodoBR search engine and also for 100 randomly selected queries. We presented a study of the user behavior in the TodoBR search engine and evaluated our suggestion method. Finally, we performed experiments to evaluate the effectiveness of our method as a query expansion technique.

4.1 Quality of suggestions

In this section we show how we have evaluated the suggestions returned by our method. This experiments were performed using a log with 2,312,586 queries from the TodoBR search engine. For these experiments we evaluated our method using minimum support equivalent to three sessions.

Table 1 shows related queries found by our method for the top 5 most popular queries at TodoBR. The translation of non English words are presented in parenthesis to make results clearer. For instance, "jogos" means "games", "jogos gratis" means "free games", and so on. The results for the top 5 queries where quite good, despite of the wrong suggestions of "sexo" related to the query "games". This occurred because "sexo" is a very common query submitted by users that also searched for "games".

Table 1. Examples of related queries.

Query	**Suggestions**
jogos (games)	fliperama (a Brazilian website about games), games jogos (games), sexo (sex), gratis jogos (free games), game
papel de parede (wallpaper)	"papel de parede" ("wall paper"), protecao tela (screen protectors), baixaki (a Brazilian website that people can download wallpapers), wallpaper, computador papel parede (computer's wallpaper)
musicas (musics)	letras musicas (music lyrics), mp3, radio (radio), mp3 musicas (musics in mp3), sexo (sex)
concursos (contest)	dirigida folha (a famous Brazilian website about public contest), concursos publicos (public contest), concurso inss (contest at inss), inss, "concursos publicos" ("public contests")
receita federal	imposto de renda (income tax), "receita federal"(Brazilian government agency responsible for tax collection), receitafederal, declaracao imposto renda (income tax declaration), fazenda ministerio (Department of Treasury)

Table 2 shows the results for an experiment using the top 95 most popular queries. For each query we evaluated the first 5, 10, 15 and 20 suggestions given by our system. Considering the first 5 suggestions, more than 90% of the results suggested by our system where correct. The judgment about the quality of the relationship between queries was performed manually by computer science undergraduate students. They were presented with the question: "Given that you searched for these keywords, would the following queries be suitable suggestions for refining your query? ".

[e] Stop words are words that are very common in a given language, such as articles and prepositions. Examples in english are the words "the" and "and".

We also performed experiments with queries randomly selected from logs. This experiment was performed to check the overall performance of our method against the complete set of queries, and not only over common queries. Using the randomly selected queries, fewer suggestions are identified (2.14 on average) since these less popular queries have fewer relationships with support above the *minsup* threshold of 3 sessions. However, the quality of the suggestions presented indicates that our method is still useful even for the general case, obtaining 93.45% of succesful suggestions.

4.2 Evaluation of user behavior

We analyzed the access and click-through logs of the TodoBR search engine to evaluate the effectiveness in a real use search experience. The system setup is very simple. After the user submits his query to the search engine, we present the top 3 queries (if available) before the query results. Figure 4 illustrates the results for the query "mp3". Notice that the suggestions given were "musica" (music), "mp3 gratis" (free mp3), and "kazaa". The user has the choice to click on an answer presented from the search engine, to click on a suggestion if he did not find a good answer, or to submit a new query.

For analyzing the user behavior, we once again generated each user session, according to the same criteria as before (t=10min). We classified each log entry into five valid actions, as follows:

- New Query: user initial query in the session.
- Suggestion Click: user clicked on a query suggestion.
- Answer Click: user clicked on one of the answers for the query.
- Query Redefinition: user reformulated his query by entering new terms.
- Give up: user gave up and left the search engine.

Figure 5 illustrates the user valid actions. We say that a user session is successful if the user reaches the "Answer Click" state, that is, the user finds some page that interested him.

Figure 6 presents an analysis of the users behavior according to the user valid actions. We analyzed only the queries that had one or more suggestions (31.73% of the total). From those, 43.8% resulted directly in a answer click (e.g., the session was successful) and 56.17% of the users were not satisfied with the initial set of presented answers. The unsatisfied users had then 3 choices: 32.9% gave up, 21.67% clicked on a suggestion and 45.40% manually redefined the query. By further analyzing the user behavior, we can reach some interesting conclusions:

- Users that click on suggestions are very likely to be successful (93.23%).

Table 2. Top 95 query suggestions.

Suggestions per query	Correct suggestions	Wrong suggestions
5	90.5%	9.5%
10	89.5%	10.5%
15	86.9%	16.1%
20	81.4%	18.6%

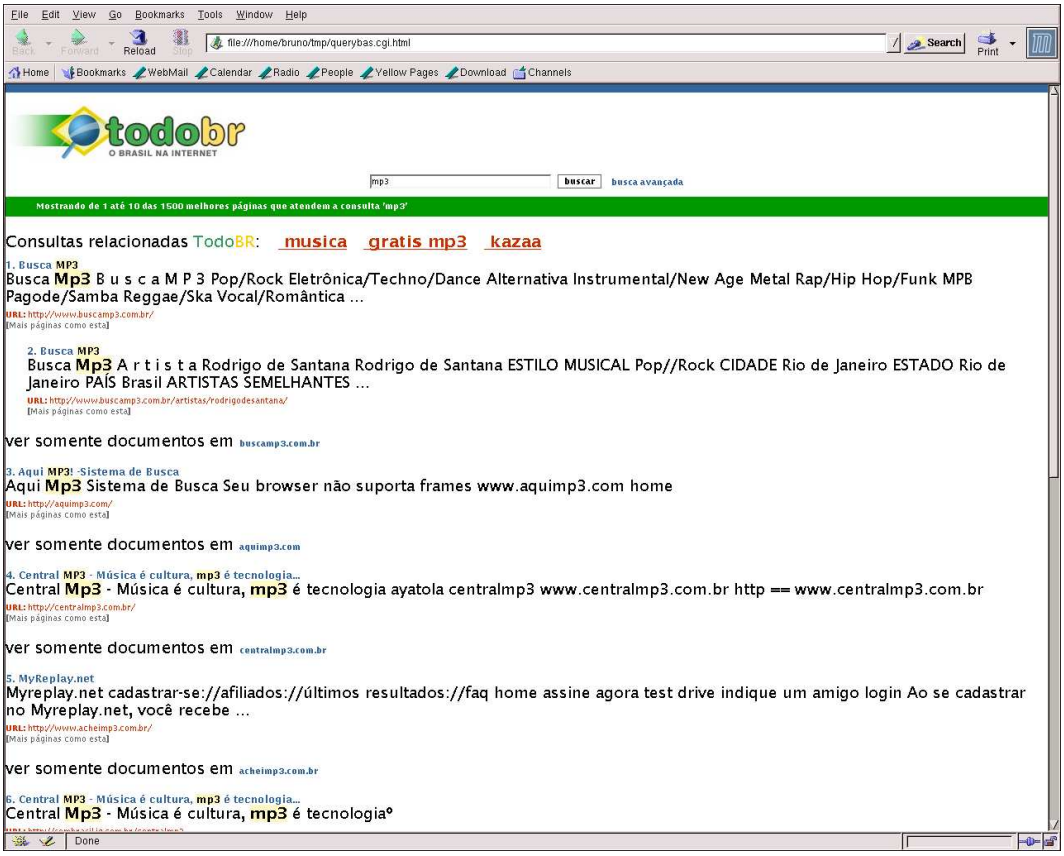

Fig. 4. The results for query "mp3" at TodoBR search engine.

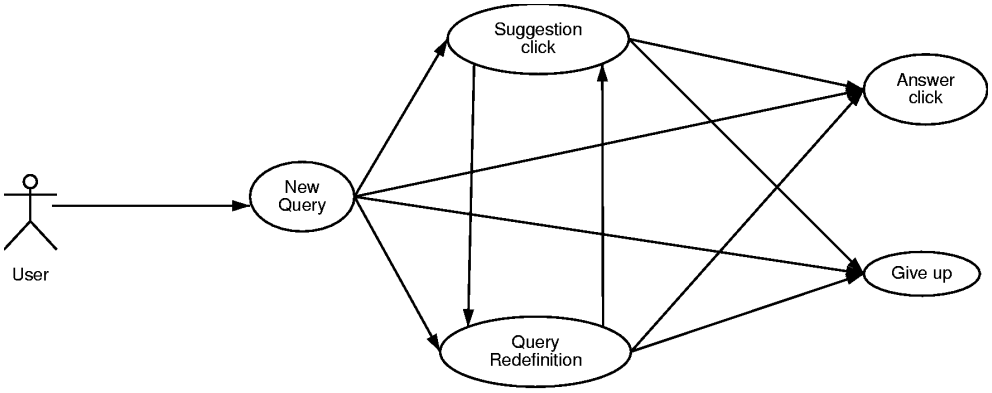

Fig. 5. User's valid actions.

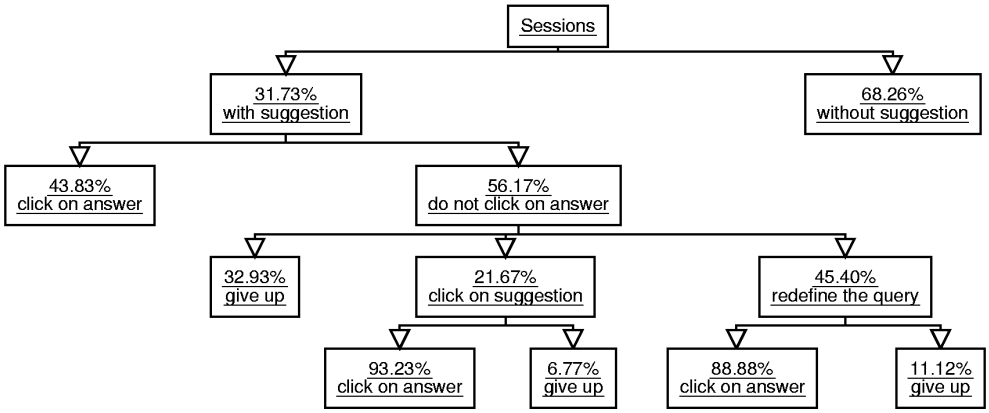

Fig. 6. Analysis of the users behavior.

- Users that click on suggestions have a higher probability of being successful than those who redefines the queries (93.23% versus 88.88%).
- There is a real need for tools that help users redefining their queries since there is a high probability that they are not happy with the initial set of answers (56.17%).

4.3 Query expansion

In this section we show how to use our suggestion system to perform query expansion based on query logs. Query expansion has long been considered as an effective way to resolve the short query and word mismatching problems. A number of query expansion methods have been proposed in traditional information retrieval [4, 13, 16, 5, 25]. However, these previous methods do not take into account the specific characteristics of web searching; in particular, of the availability of large amount of user interaction information recorded in the Web query logs. In our method, the new query is obtained by combining the original query and its suggestions in a disjunctive way, i.e., with the connective "OR".

Now we show the retrieval performance of the log-based query expansion with the baseline (without query expansion). Interpolated 11-point average precision is employed as the main metric of retrieval performance. The results of our query expansion approach for a range of suggestions from 1 to 5 can be seen in Table 3 and Figure 7. We submitted all queries to the search engine Google. We see that our log-based query expansion performs well on the experiments. It brings an improvement ranging from 5.47% to 23.16% in average precision over the baseline method. It indicates that query expansion is extremely important for short queries. The best results were found when we expand the original query with three suggestions. We chose only the top-10 answers to validate our approach. These small number of selected answers leads to a range of empty values in the recall x precision curve.

Generally, log-based query expansion selects expansion terms from a relatively narrower but more concentrated area. In contrast, document-based query expansion techniques, which selects terms in the top-ranked retrieved documents, is more likely to add some irrelevant

Table 3. Recall x Precision for Query Expansion

Recall (%)	**Precision(%)**					
	Orig.	*1*	*2*	*3*	*4*	*5*
0	87.69	90.03	95.04	97.24	99.34	94.36
10	78.02	75.19	87.36	91.66	84.18	84.71
20	61.17	60.35	74.01	83.16	74.70	72.87
30	22.63	36.14	32.28	39.88	38.94	50.41
40	7.37	8.25	7.89	4.74	4.73	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00
70	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00
90	0.00	0.00	0.00	0.00	0.00	0.00
100	0.00	0.00	0.00	0.00	0.00	0.00
Average	23.18	24.45	26.82	28.55	27.18	27.27
Improvement		5.47	15.70	23.16	17.25	17.64

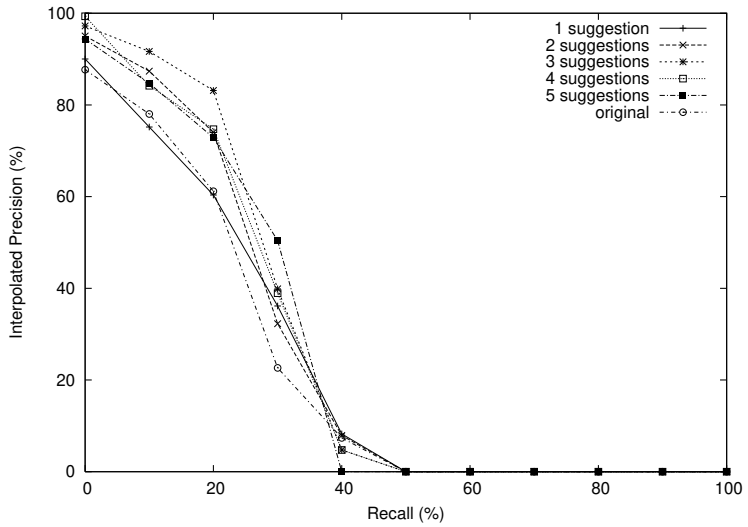

Fig. 7. Recall x Precision for Expanded Queries

terms into the original query, thus introducing undesirable side effects on retrieval performance.

5 Conclusions and Future Work

We have shown a method for proposing related queries based on the application of association rules over search engine query logs. The method proposed is simple, has low computational cost and presents good results. The experiments presented show the practical usefulness of the approach. There is a high probability that users who click on suggestions find relevant documents.

We have also experimented our method as input to a query expansion technique. This

initial experiment has shown that using an OR operator between the original and some related queries, we obtained an improvement in the average precision of the answers of the Google search engine.

As future work we are planning to expand our method by studying the possibility of new combinations among original and related queries. We will compare our query expansion approach with other approaches based on document content. We are also studying the possibility of combining the information extracted from the logs with information extracted from the Web documents to derive a new suggestion method. Finally, we plan to extend our method for mining association rules to use itemsets with more elements.

Acknowledgments

The authors acknowledge the support by Akwan Information Technologies in providing the log files for this research. This work was supported in part by GERINDO Project–grant MCT/CNPq/CT-INFO 552.087/02-5, CYTED VII.19 RIBIDI Project and by CNPq Grant 520.916/94-8 (Nivio Ziviani), CNPq scholarship 141.269/02-2 (Bruno Pôssas).

References

1. R. Agrawal, T. Imielinski, and A. Swami, *Mining association rules between sets of items in large databases*, in Proceedings of the ACM SIGMOD International Conference Management of Data, Washington, D.C., May 1993, pp. 207-216.
2. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo, *Fast discovery of association rules*, in Advances in Knowledge Discovery and Data Mining, San Jose, CA, 1996, AAAI/MIT Press, pp. 307-328.
3. R. Agrawal and R. Srikant, *Fast algorithms for mining association rules*, in The 20th International Conference on Very Large Data Bases, Santiago, Chile, September 1994, pp. 487-499.
4. R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley, Essex, England, 1999. 513 pages.
5. C. Buckley, G. Salton, J. Allan, and A. Singhal, *Automatic query expansion using SMART: TREC 3*, in Text REtrieval Conference, 1994, pp. 0–.
6. P. Calado, B. Ribeiro-Neto, N. Ziviani, E. Moura, and I. Silva, *Local versus global link information in the web*, ACM Transactions on Information Systems (TOIS), 21 (2003), pp. 42–63.
7. R. Cooley, B. Mobasher, and J. Srivastava:, *Data preparation for mining world wide web browsing patterns*, Knowledge and Information Systems, 1 (1999), pp. 5–32.
8. H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, *Probabilistic query expansion using query logs*, in Proceedings of the eleventh international conference on World Wide Web, ACM Press, 2002, pp. 325–332.
9. A. Geyer-Schulz and M. Hashler, *Evaluation of recommender algorithms for an internet information broker based on simple association rules and on the repeat-buying theory*, in Proceedings of WEBKDD'2002, Edmonton, Canada, July 2002, pp. 100–114.
10. N. S. Glance, *Community search assistant*, in Proceedings of the 6th international conference on Intelligent user interfaces, ACM Press, 2001, pp. 91–96.
11. D. Harman, *Relevance feedback revisited*, in Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press, 1992, pp. 1–10.
12. C.-K. Huang, L.-F. Chien, and Y.-J. Oyang, *Relevant term suggestion in interactive web search based on contextual information in query session logs*, J. Am. Soc. Inf. Sci. Technol., 54 (2003), pp. 638–649.
13. j. Xu and W. B. Croft, *Improving the effectiveness of information retrieval with the local context*

analysis, ACM Transaction of Information Systems, 18 (2000), pp. 79–112.
14. B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic, *Real life information retrieval: a study of user queries on the web*, ACM SIGIR Forum, 32 (1998), pp. 5–17.
15. W. Lin, S. Alvarez, and C. Ruiz, *Efficient adaptive-support association rule mining for recommended systems*, Data mining and knoledge discovery, 6 (2002), pp. 83–105.
16. M. Mitra, A. Singhal, and C. Buckley, *Improving automatic query expansion*, in Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press, 1998, pp. 206–214.
17. J. Park, M. Chen, and P. Yu, *An effective hash based algorithm for mining associative rules*, in Proceedings of the ACM SIGMOD International Conference on Management of Data, San Jose, CA, May 1995, pp. 175-186.
18. B. Pôssas, W. Meira, M. Carvalho, and R. Resende, *Using quantitative information for efficient association rule generation*, in ACM Sigmod Record, December 2000.
19. J. J. Rochio, *Relevance feedback in information retrieval*, The SMART retrieval system - experiments in automatic document processing, (1971).
20. N. C. M. Ross, *End user searching on the internet: an analysis of term pair topics submitted to the excite search engine*, J. Am. Soc. Inf. Sci., 51 (2000), pp. 949–958.
21. A. Savasere, E. Omiecinski, and S. Navathe, *An efficient algorithm for mining association rules in large databases*, in The 2lst International Conference on Very Large Data Bases, Zurich, Switzerland, September 1995, pp. 432-444.
22. C. Silverstein, M. Henzinger, H. Marais, and M. Moricz, *Analysis of a very large altavista query log*, Tech. Rep. 1998-014, Digital SRC, 1998. http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1998-014.html.
23. J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, *Clustering user queries of a search engine*, in Proceedings of the tenth international conference on World Wide Web, ACM Press, 2001, pp. 162–168.
24. I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*, Morgan Kaufmann Publishers, 2nd ed., 1999.
25. J. Xu and W. B. Croft, *Query expansion using local and global document analysis*, in Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press, 1996, pp. 4–11.
26. M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, *New algorithms for fast discovery of association rules*, in Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, August 1997, pp. 283-286.
27. M. J. Zaki, *Generating non-redundant association rules*, in 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, August 2000, pp. 34-43.