Analyzing the behavior of professional video searchers using RAI query logs

Claudio Carpineto, Giovanni Romano, Andrea Bernardini
Fondazione Ugo Bordoni
Rome, Italy
{carpinet, romano, abernardini}@fub.it

Abstract

A large number of studies have investigated the query logs of Web search engines, but there is a lack of analogous studies for multimedia database management systems (MDBMSs) used by professional searchers. In this paper we perform an extensive analysis of the query logs of the RAI multimedia catalogue, both at the query level and at the session level. Based on the observation that a large proportion of the queries returned zero or, conversely, too many hits, we identified three query reformulation strategies to reduce or enlarge the set of results. Our study indicates that the desire of controlling the amount of output may have a relatively limited (moderate-to-little) impact on the user's behavior, while at the same time some counter-intuitive findings suggest a suboptimal utilization of the system. The findings are useful for MDBMS developers and for trainers of professional searchers to improve the performance of interactive searches, and for researchers to conduct further work.

1. Introduction

By turning to the vast stream of log data that users generate while looking for information, one can gain better insights into user's searching behavior in an attempt to improve the design of search system interfaces. Query log analysis has attracted much attention recently, with a number of dedicated studies. However, this research line has primarily focused on search engine users. Earlier studies of large query logs collected in multimedia database management systems (MDBMSs) involving professional users have been rare; e.g., [1], [2].

Previous research has shown that an analysis of query logs at the level of the single queries is useful but it is limited by the brief duration of the interaction and by the lack of contextual information. In order to capture a whole session of the interaction between user and system, we should investigate multiple queries. A session can be defined as "a series of interactions by the user toward addressing a single information need" [3]. The notion of session is thus tightly coupled with the effort made by the user to modify a previous search query in hope of retrieving better results. While it is possible to classify a query as an initial query or a reformulation, it is much more difficult to understand how users perform query reformulation. This issue has been addressed in [4] and [5], in the context of search engine query logs.

A better understanding of the interactive strategies employed by the users to find the wanted information is desirable not only for search engines but also for MDBMSs. However, the query logs of a MDBMS are different from those of a search engine because the query language is structured and the user may specify a query in a much richer way than using just a set of keywords. In this paper we adapt and extend the earlier techniques for detecting session boundaries and modeling query reformulation strategies to the new requirements posed by a MDBMS. In particular, we identify three major query reformulation strategies [6] to modify the size of the result set: specialization, generalization, re-focusing, and repetition.

Using the novel techniques, we analyze the query logs collected in 2010 in the multimedia catalogue “Teche RAI”. We report a number of findings that partly confirm earlier results obtained for Web searches and party contradict our expectations. On one hand, we noticed that the majority of queries were simply formed by specifying one or more keywords, that the similarity between queries was low, and that sessions usually consisted of few queries and ended with a query returning few results. On the other hand, we found that the preferred user strategy was query re-focusing (even when the result set was very small or very large), and that the specialization and

generalization strategies were often used in an inconsistent manner. The latter findings are more puzzling. The suboptimal use of the query reformulation strategies suggests that the retrieval logic employed by the searchers was different from that actually implemented in the MDBMS.

The remaining of the paper has the following organization. We first briefly describe the RAI video search systems and the query log data. Then we present the results of our analysis at the query level. Preceded by a discussion about the criterion used to identify sessions, we next present the results of our analysis at the session level. The last part of the paper is about the query reformulation strategies. We finally offer the main conclusions of our study.

2. The Teche RAI video search system

RAI is the Italian public broadcaster. "Catalogo Multimediale Teche" is a MDBMS used by RAI professionals to retrieve video fragments from a vast archive of RAI transmissions. The catalogue is expanded each year by digitally storing all Rai television and radio broadcasts and by gradually recovering past material, as early as 1954. At the end of 2010, a total of 2,009,306 hours were made available for consultation to 10,500 users.

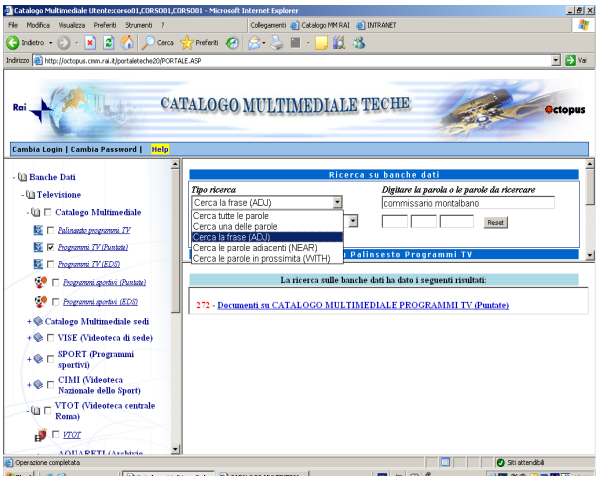

Figure 1. A screenshot of Catalogo Multimediale Teche, with a phrase search option selected.

RAI multimedia documents have structured descriptions and users may express their queries specifying the values for a number of attributes; e.g., archive, date, hour, etc. The user can also search the textual descriptions of various document fields for sets of words, phrases, proximity words, and many other text constructs. All the documents (usually video fragments) that satisfy all the conditions of a query are shown to the user as a list. The results are not ordered by relevance. As an illustration, in Figure 1 we show a screenshot of the system where the user has chosen the phrase search option ("Cerca la frase", in Italian), typing "commissario montalbano" as the corresponding value.

3. Query log data

The data used in our study contained all queries collected in 2010. After some data cleaning and formatting, we finally identified 2,099,449 queries, each being represented as a set of attribute-value pairs. Most attributes were in fact search conditions, each value being the argument (provided by the user) of the corresponding search condition. We checked that the query log contained 424 distinct types of attributes. An important attribute contained in the query log was the number of results satisfying each query. As an example, consider the following query (extracted from the corresponding XML version):

40 11 2010-01-01 17:31:45 1 69 34 2 318 venezuela

The query, identified by the number 40, was made on January 1, 2010 at 17 31' 45", searching the regional TV archives for the word "venezuela" in the name of TV programs, and the system returned 69 results.

4. Analysis at the query level

Users made a limited use of the features provided by the rich query language. We found that 98% of queries were simple searches based on sets of words, sometimes with an additional constraint on the date (i.e., for about 10% of queries). The total number of distinct words was 284,351, including typos. The five most frequent words, together with their number of occurrences, were: "di" ("of") - 36,897, "grezzo" ("raw") - 28,539, "tg2" (21,375), "la" ('the") - 18,835, "e" ("and") - 16,579. The analysis at the query level also revealed that the same terms usually did not occur in multiple queries. Only 30,991 words were present in more than 10 queries.

5. Detecting boundaries for MDBMS sessions

In order to better understand the user's searching behavior, we need to group multiple queries in a full

search session. However, sessions can be identified only in an approximate manner. Most studies on Web query logs rely on a temporal cutoff, assuming that a certain time interval of inactivity by a user (typically thirty minutes) constitutes the end of a session. The advantage of this approach is its simplicity, but a recent study has shown that any predetermined time threshold is arbitrary and its ability to correctly identify the boundaries of a session is comparable to that of a randomly chosen time [7].

A more effective technique is based on the similarity between consecutive queries. The most common approach is to assume that two consecutive queries by the same user are part of the same session when they contain at least one word in common. Consequently, the beginning of a new session is marked by the presence of a query that has no words in common with the previous one. It is also possible to assess the similarity between two queries using more sophisticated approaches than strict word matching, such as using syntactic or semantic distances between the full-text query strings, but their advantages are still not clear.

The criterion based on words in common between consecutive queries works quite well for Web searches but it may be too strict for MDBMSs. A MDBMS user can specify a query using a number of conditions that do not require user keywords. The focus of the search can thus be more easily maintained even providing altogether different search words. For instance, a user can search within a series of documentaries entitled "the sea world" those fragments dealing with dolphins, whales, etc. These different queries likely refer to the same search session.

Based on these observations, our strategy was to add a new query to the current session when the query had at least one word in common to at least one query in the current session. In Table 1 we show an example, in which the last query of the session contains a single word (i.e., the name of the TV program) that appeared only in the first query of the session. Note that all query words occurred within a same search condition (i.e., "search for the following words"), while other attribute-value pairs not shown in Table 1 (e.g., user id, archive) stayed the same across all queries.

Table 1. Example of session about the film director "Mario Monicelli" interviewed at the TV program "Annozero".

Date and hour	# of results	Query words
2010-12-13 15:29:56	0	monicelli annozero
2010-12-13 15:30:11	1431	monicelli
2010-12-13 15:30:26	157	monicelli interviste
2010-12-13 15:31:48	1431	monicelli
2010-12-13 15:33:04	1076	mario monicelli
2010-12-13 15:34:16	694	annozero

6. Analysis at the session level

Using the method described above, we detected 882803 sessions. Sessions with few queries were of course much more numerous than those having many queries. A good part of the sessions, namely 271,173 (i.e., 30.71%), contained only one query. Considering the sessions that contain a maximum of 10 queries, we covered the 84% of the sample. The longer session contained 794 queries.

In Figure 2 we plot the number of the sessions as a function of the session length, in the range between 1 and 100. These sessions cover a total of 99.92% of the sample. This pattern is similar to a power-law distribution, albeit with some singular point (especially when the length is equal to 7).

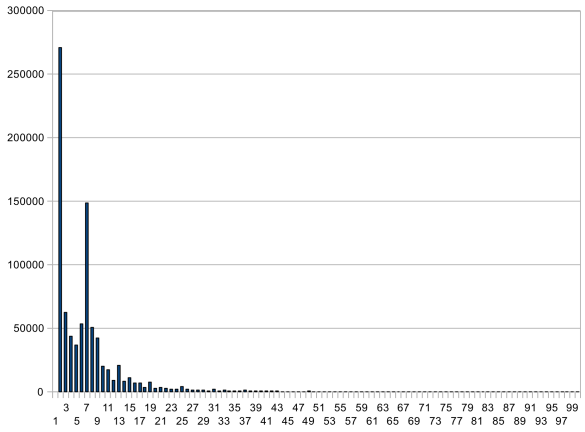

Figure 2. Distribution of sessions over the number of queries contained in a session.

A related aspect is the temporal length of sessions. This computation is necessarily approximated, because the time spent to analyze the results returned by the last query of a session is not known. Using the number of seconds elapsed between the first and the last question, we found that the relationship between number of

sessions and duration is represented by a power-law, similar to the behavior found using the number of queries as length. Figure 3 shows this behavior for the interval between zero and 100 seconds. The sessions that last up to 100 seconds covered the 80.39% of the sample. The longest session lasted 82307 seconds (i.e., about 23 hours). The presence of very long sessions indicates that some users continued to search the same subject after resuming their work.

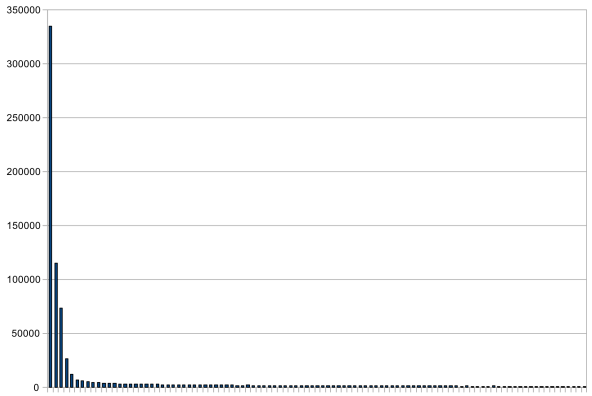

Figure 3. Distribution of sessions over the duration. The x-axis varies from 0 to 100 seconds.

Another interesting issue to be analyzed carefully is the relationship between the user's behavior and the number of results associated with the single queries within each session. When the system returns a large number of results, it is not possible to examine all of them. Conversely, if the system finds few or no documents that satisfy the query, the user has to express his information need using different words or attributes. It is conceivable that the user starts with a rather general query – returning many results – and then works toward gradually refining its scope and the size of the associated set of results. This hypothesis can be explored by examining the log data.

Considering the 337,920 sessions with at least two queries, we found that, on average, the first query returned 1,650 results and the last query 534 results. While there is a sharp decrease in the number of results passing from the first to the last query, the numbers are still high, on an absolute scale. This is due, in part, to the presence of a few queries with a very large number of results (queries such as "tg1" or "tg2" return millions of documents). To overcome this problem, it is more convenient to consider the median. The median of the number of documents returned by the first and the last queries are, respectively, 13 and 1. This is an indication that many sessions end with zero results (mean = 1) or, conversely, with many results (mean = 534).

To better understand this phenomenon, in Figure 4 we plot the number of sessions (including those containing only one query) as a function of the number of results returned by the last query of each session. The Y-axis is in logarithmic scale and the final value is the total number of sessions ending with more than 1000 results. The curve clearly shows that the vast majority of sessions end with few results. For instance, the sessions ending with at most 10 results are 562,074 (i.e., 63%). A more surprising finding is that 218,841 sessions (about 25% of the sample) end with zero results. Of these, about 60,000 sessions contain only one query. A possible explanation for the sessions with zero final results is that the user reckons that he cannot retrieve more relevant documents (or that he cannot retrieve any relevant document at all). Figure 4 also shows that some tens of thousands of sessions end with more than 1,000 results. Here the user may have decided that it is too difficult to further reduce the set of results, or he may have retrieved the sought items in the first results returned by the system.

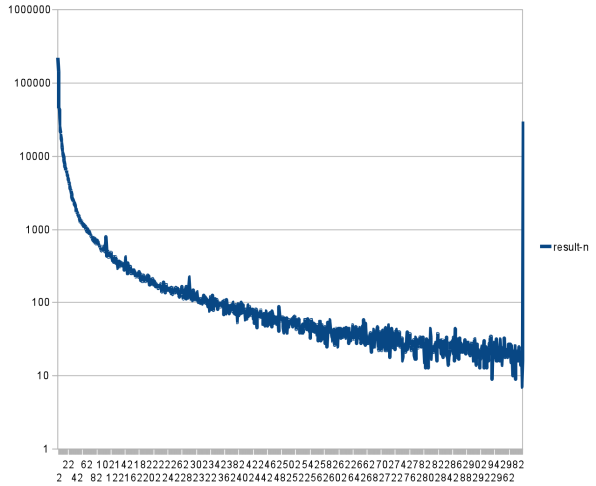

Figure 4. Number of sessions as a function of the number of results returned by the last query of each session. The Y axis is in logarithmic scale. The final value is the total number of sessions ending with more than 1000 results.

7. Query reformulation strategies

In this section we study how the users performed query reformulation. Identifying query reformulations is equivalent to detecting session boundaries: the first

query of a session is a new query, while the following queries are reformulations. Each query within a session (except for the first query) can be classified in one of the following classes with respect to the preceding query:

- Specialization, i.e., a more specific query that returns a subset of results.
- Generalization, i.e., a more general query that returns a superset of results.
- Re-focusing, i.e., a query neither more specific nor more general that returns neither a subset nor a superset of results.

For instance, the classes of the reformulated queries in the example session in Table 1 are: generalization, specialization, generalization, specialization, and re-focusing. Each of these strategies can be assessed by comparing the current query to the preceding one. The precise definition of the rules used to classify each question is the following.

- Specialization. The query can be obtained by adding conditions and/or adding terms within an existing textual condition.
- Generalization. The query can be derived by removing conditions and/or removing terms within an existing textual condition.
- Re-focusing. The query can be derived by adding conditions and/or adding terms within an existing textual condition, and at the same time by removing conditions and/or removing terms within an existing textual condition.

In Table 2 we show the distribution of the reformulation strategies along with their sub-strategies across the set of 1,228,395 queries to which they can be applied; i.e. the total number of queries (2,099,449) minus the number of single-query sessions (871,054). Note that the sum of the sub-strategies is greater than the corresponding strategy because multiple sub-strategies may be applied simultaneously. A special case of query re-focusing is the repetition of the same query, where the user obtains an identical set of results. As this event – termed Replication – occurred frequently (i.e., 30% of times), we treated it separately in Table 2. We do not have a convincing explanation for this phenomenon. However, we noticed that a large fraction of consecutive identical queries had nearly the same time stamps, which may be attributed to an involuntary double click. The other replications may be related to the poor response times of the system.

In order to understand how the choice of using one reformulation strategy or another was related to the size of the result set, in Figure 5 we plot the number of queries classified in each class as a function of the number of results returned by the preceding query. (from 1 to 50).

Table 2. Distribution of strategies and sub strategies across queries.

Strategies	# of queries (%)	Addition of conds (%)	Addition of words (%)	Removal of conds (%)	Removal of words (%)
Spec.	10.9	8.56	3.38		
Gen.	11.11			7.94	4.17
Re-foc.	47.3	12.1	9.8	14.05	13.75
Rep.	30.69				

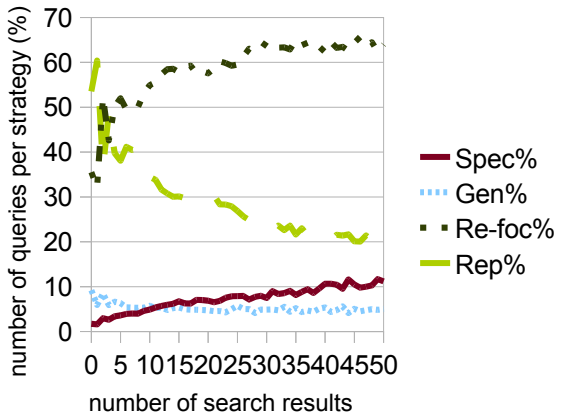

Figure 5. Number of queries (in percentage) of each reformulation strategy (including query replication, denoted "eq"), as a function of the number of results returned by the preceding query.

The first observation is that Generalization and Specialization were less frequent than Re-focusing and Replication across all the sizes of interest (for the chosen range). The curve of Replication declines sharply, compensated by an equivalent increase of Re-focusing. The fact that Re-focusing grows as the number of results increases – in the range 1-50 – can be explained considering that these queries contain neither too many nor too few conditions, and therefore it may be technically easier to add and simultaneously remove some conditions.

The second remark is that, as expected, Generalization and Specialization are, respectively, inversely and directly proportional to the number of results returned by the query to be reformulated. However, Figure 5 also shows that many users applied

Specialization when the number of results was very small (or even equal to zero), which of course is meaningless because they will obtain – by definition – an even smaller set. Likewise, many users kept applying Generalization even when the set of results grew large.

This behavior was probably due to a mismatch between the retrieval logic of the system (where the query conditions are ANDed together) and that of the user. One possible interpretation is that some users thought that adding new conditions and terms would favor retrieval of more diverse results (thus making the query more general), while removing conditions and terms would help to get rid of irrelevant results matching only part of the query (thus making the query more specific). In other words, our hypothesis is that many users implicitly believed that the query conditions would be ORed together, which led to a suboptimal use of the system.

8. Conclusions

The main conclusions of our analysis of the RAI query logs are the following.

- The rich query language of the MDBMS was used only to a very limited extent, because the vast majority of queries consisted of simple searches for a set of words.

- Most sessions had a short duration and were formed by few queries. The 30% of sessions contained only one query.

- Most sessions ended with a query returning few results. The 25% of sessions ended with a query with empty results.

- The desire to control the amount of output had a low-to-moderate effect on how the users modified their queries. Most queries can be seen as an attempt to re-focusing the search.

- In 30% of cases, the new query of a multi-query session was equal to the preceding query.

- The specialization and generalization strategies of query reformulation were used in an inconsistent manner by a minority of users.

These findings may be useful for the development of tools for assistance in query formulation and, more in general, for the design of better MDBMS browsing interfaces. They may also be of some value to trainers of professional searchers to improve their productivity.

The next step of this research is to extend the analysis of RAI query logs to include not only the number of hits associated with each query but also the actual results returned by the system and the actions made by the users while browsing through the list of results.

9. Acknowledgments

This research is supported by POR FESR Lazio 2007-2013 funding scheme within IRMA (Intelligent Retrieval in Multimedia Archives) project. We would like to thank RAI for providing the query log data.

10. References

[1] S. Mongy. “A study on video viewing behavior: application to movie trailer miner”. IJPEDS 22(3), pp. 118-125, 2007.

[2] P. Huntington, D. Nichols, and H. R. Jamali. “Employing log metrics to evaluate search behaviour and success: case study BBC search engine”, JIS 33(5), pp. 1-21, 2007.

[3] B. J. Jansen, A. Spink, C. Blakely, and S. Koshman. “Defining a session on Web search engines: Research Articles”, J. Am. Soc. Inf. Sci. Technol. 58, 6 (April 2007), pp. 862-871, 2007.

[4] T. Liang-Chun, J. Tjondronegoro, D. Wirawan and A. H. Spink, “Analyzing web multimedia query reformulation behavior”, In Proceedings of the 14th Australasian Document Computing Symposium, CSIRO, University of New South Wales, Sydney, pp. 118-125, 2009

[5] J. Huang and E. N. Efthimiadis.. “Analyzing and evaluating query reformulation strategies in web search logs”, In Proceedings of the 18th ACM conference on Information and knowledge management (CIKM '09). ACM, New York, NY, USA, pp. 77-86, 2009

[6] T. Lau, E. Horvitz. “Patterns of search: analyzing and modeling Web query refinement”, In Proceedings of the seventh international conference on User modeling (UM '99), Judy Kay (Ed.). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 119-128, 1999.

[7] R. Jones, K.L. Klinkner, “Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs”, In CIKM ‘08, pp. 699-708. 2008