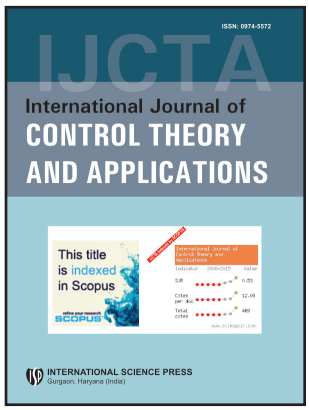

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 16 • 2017

Predict Keyword Based Search Process using Semantic Method

B. Bazeer Ahamed[a] and T. Ramkumar[b]

[a]*Assistant Professor in CSE department, MIET Engineering college Trichirappalli, India*
[b]*Associate Professor in the School of Information Technology and Engineering, VIT University, Vellore - 632 014 Tamil Nadu India*

Abstract: Sprouting of web day by day, people converse in various search engine for the dominant results. The accountability of web site provider can give relevant information for the satisfaction of web browser. In the current scenario of web search process, there are various ranking algorithms are present for getting the desired result. We propose to analyses the characteristics of difficult keyword queries over various databases and popular keyword search ranking methods. The difficulty of a query based on the difference between the ranking of the same query over the original and noisy versions of the same database. The propose technique is to improve efficient ranking and the result is much more accurate with minimum amount of time.

Keywords: Semantic search; semantic relations; Web Usage Mining, Related pages; Prediction.

1. INTRODUCTION

Information leads to power, success, and sophisticated technologies such as computers, satellites, etc., tremendous amounts of information have been collected. Initially, with the advent of computers and means for mass digital storage, started collecting and storing all sorts of data. These massive collections of data stored on disparate structures very rapidly became over whelming[3]. This initial chaos has led to the creation of structured databases and database management systems. The efficient database management systems have been very important assets for management of a large corpus of data especially for efficient retrieval of particular information from a large collection whenever needed.

The proliferation of database management systems has also contributed to recent massive gathering of all sorts of information. Today, there is a need to handle more information such as business transactions and scientific data, to satellite pictures, text reports and military intelligence. Information retrieval is simply not enough anymore for decision-making. These needs are automatic summarization of data, extraction of the "essence" of information stored, and the discovery of patterns in raw data.

Keyword search is the de facto Information retrieval mechanism for data on the World Wide Web. It also proves to be an effective mechanism for querying Semi-structured and structured data, because of its user-friendly query inter-Face. In this method focus on keyword search problems for xml documents (Semi-structured data),

relational databases (structured data), and all kinds of Schema-free graph data Keyword queries on databases provide easy access to data, but often suffer from low ranking quality, i.e., low precision and/or recall, as shown in recent benchmarks. It would be useful to identify queries that are likely to have low ranking quality to improve the user satisfaction[4].

For instance, the system may suggest to the user alternative queries for such hard queries. So, analyze the characteristics of hard queries and propose a novel framework to measure the degree of difficulty for a keyword query over a database, considering both the structure and the content of the database and the query results. Evaluate our query difficulty prediction model against two effectiveness benchmarks for popular keyword search ranking methods. The KQI is considered to be important if it is satisfies the following contributions:

- Predicting the degree of the difficulty for queries over database
- Structured robustness (SR) score measure the difficulty of a query based on the differences between the rankings of the same query over the original and noisy versions of the same database.
- Algorithm computes the SR score. And parameters to tune its performance
- Approximation algorithms to estimate the SR score, given that such a measure is only useful

when it computed with a small time overhead compared to the query execution time. Ranking algorithms consider how often keywords appear in a document (frequency). They also measure keywords in relation to each other within a document (proximity). Another measure considers the location of keywords in a document. Keywords occurring at the beginning of a page, in the titles of pages, and in the URLs of the pages, are all given more 'weight' as relevancy is determined.

Each search engine determines the relevance of a page as it relates to a query by using a ranking algorithm. The ranking algorithm is a computerized formula designed to match highly relevant pages with a user's query. In general, search engines use a combination of factors that always include keyword frequency and page popularity. If a query is well formed, the results, while imperfect, often satisfy the searcher.

2. RELATED WORK

A. A Framework to Improve Keyword Search over Entity Databases

Keyword search over entity databases (e.g., product, movie databases) is an important problem. Current techniques for keyword search on databases may often return incomplete and imprecise results [1]. On the one hand, they either require that relevant entities contain all (or most) of the query keywords, or that relevant entities and the query keywords occur together in several documents from a known collection. The above technique creates a framework that can improve an existing search interface by translating a keyword query to a structured query. Specifically, leverage the keyword to attribute value associations discovered in the results returned by the original search interface [3].

Differential query pair (DQP) approach uses statistical difference Aggregated over several selectively chosen query pairs to find the mappings from keywords to predicates. In the process first define differential query pairs, and then show how aggregation over multiple differential query pairs allows us to derive accurate mappings. Baseline keyword search interface over an entity database, to map keywords in a query to predicates or ordering clauses. Based on the above framework we have few advantages as Validating this approach using experiments conducted on multiple search interface over the real data sets, and concluded that keyword++ is a viable alternative to answering keyword queries and The problem of leveraging existing keyword search interface to derive keyword to predicate mappings, which can then be used to construct SQL for robust keyword query answering will be disadvantage.

B. A Probabilistic Framework for Query Performance Prediction

The query-performance prediction task is stated as estimating the effectiveness of a ranking induced by retrieval method *m* over a corpus of documents d in response to query *q* in lack of relevance judgments [4] .The result list and its ranking, rather than address the entire corpus ranking, as this is also the focus of the most commonly used evaluation measures (e.g., average precision, precision at top ranks). The framework has advantage as Post-retrieval predictors simply differ by the choice of the pseudo (in) effective ranking that serves for reference, and/or the inter ranking similarity measure used where as disadvantages are common formal grounds to various previously proposed prediction methods that might seem to rely on completely different principles and hypotheses, Providing new insights about commonly used prediction methods such as Clarity and the connections between them and Giving rise, based on formal arguments, to new prediction approaches that were empirically shown to improve over the state-of-the-art. Integrating various prediction types that emerged in our framework using additional approaches [5].

C. A Unified Framework for Post-Retrieval Query-Performance Prediction

The query-performance prediction task is estimating the effectiveness of a search performed in response to a query in lack of relevance judgments. Post-retrieval predictors analyze the result list of top-retrieved documents. While many of these previously proposed predictors are supposedly based on different principles, and show that they can actually be derived from a novel unified prediction framework of propose[4]. The framework is based on using a pseudo effective and/or ineffective ranking as reference comparisons to the ranking at hand, the quality of which they want to predict. Empirical exploration provides support to the underlying principles, and potential merits, of our framework. A (simple) novel unified post-retrieval prediction framework that can be used to derive many previously proposed post-retrieval predictors that are supposedly based on completely different principles [7]. Query feedback. In the query feedback (QF) predictor, a query model is induced from L[k] M and is used to rank the entire corpus. This approach has Post-retrieval predictors analyze the result list of top-retrieved documents as advantage where as a novel unified framework for post-retrieval query-performance prediction which is used for deriving previously proposed predictors that are supposedly based on completely different principles [6].

3. EXPERIMENTAL RESULT

Structured Robustness (SR) score, measures the difficulty of a query based on the differences between the rankings of the same query over the original and noisy (corrupted) versions of the same database, where the noise spans on both the content and the structure of the result entities. Structured Robustness Algorithm (SR Algorithm)[8], which computes the exact SR score, based on the top K result entities. Each ranking algorithm uses some statistics about query terms or attributes values over the whole content of DB. Some examples of such statistics are the number of occurrences of a query term in all attributes values of the DB or total number of attribute values in each attribute and entity set. SR Algorithm increases the query processing time considerably [6].

Approximation algorithms to improve the efficiency of SR Algorithm. Our methods are independent of the underlying ranking algorithm.

Internet search engines have popularized keyword based search [2]. Users submit keywords to the search engine and a ranked list of documents is returned to the user. In this system analyze the characteristics of difficult keyword queries over databases and propose a novel method to detect such queries and take advantage of the structure of the data to gain insight about the degree of the difficulty of a query given the database. It is implemented some of the most popular and representative algorithms for key-word search on databases and

used them to evaluate our techniques on both the INEX and SemSearch benchmarks. The results show that our method predicts the degree of the difficulty of a query efficiently and effectively.

A. Structured Data

It is organized in a way that makes it easy for different people through it to find the topics and the level of detail that are of interest to them. Structured data analysis is the statistical data analysis of structured data. This can arise either in the form of an a priori structure such as multiple-choice questionnaires or in situations with the need to search for structure that fits the given data, either exactly or approximately [9]. This structure can then be used for making comparisons, predictions, manipulations. Structured data refers to information with a high degree of organization, such that inclusion in a relational database is seamless and readily searchable by simple, straightforward search engine algorithms or other search operations

$$\mathrm{TC}a,b=\frac{\sum_{i=1}^{\mathrm{N}}\frac{\mathrm{T}i}{\mathrm{T}ab}\times\frac{fa(k)}{fb(k)}}{\sum_{i=1}^{\mathrm{N}}\frac{\mathrm{T}i}{\mathrm{T}ab}} \tag{1}$$

where, Ti is the total time duration of the ith session that contain both the pages a and b and Tab is difference between requested time of page a and page b in the session. The value of $f(k)$ is the position of the page in the session. The time connectivity measure is normalized to hold values between 0 and 1.

B. Hard Query

Queries results are generated by accessing relevant database data and manipulating it in a way that yields the requested information. Since database structures are complex, in most cases, and especially for not-very-simple queries, the needed data for a query can be collected from a database by accessing it in different ways, through different data-structures, and in different orders. Each different way typically requires different processing time[11]. Processing times of the same query may have large variance, from a fraction of a second to hours, depending on the way selected. The purpose of query optimization, which is an automated process, is to find the way to process a given query in minimum time.

$$\mathrm{FC}a,b=\frac{\mathrm{N}ab}{\max\left\{\mathrm{N}a,\mathrm{N}b\right\}} \tag{2}$$

where Nab is the number of sessions containing both page a and b. Na and Nb are number of session containing only page a and page b.

C. Sem Search

Mapping between keywords and formal concepts is a common pattern appearing in semantic search .Semantic search is an application of the Semantic Web to search.[10] Search is both one of the most popular applications on the Web and an application with significant room for improvement and believe that the addition of explicit semantics can improve search. Semantic Search attempts to augment and improve traditional search results (based on Information Retrieval technology) by using data from the Semantic Web. Semantic search is a data searching technique in a which a search query aims to not only find keywords, but to determine the intent and contextual meaning of the words a person is using for search. Semantic search provides more meaningful search results by evaluating and understanding the search phrase and finding the most relevant results in a website, database or any other data repository.

$$W_{a,b} = \frac{2 \times TC_{ab} \times FC_{ab}}{TC_{ab} + FC_{ab}} \quad (3)$$

D. Forming pseudo-document based on URL representations

In order to obtain the feature representation we propose an optimization method to combine both clicked and unclicked URLs

Let F *fs* be the feature representation and F *fs*(μ) be the value for the term. Let $FF_{fs}(\mu)(m = 1, 2, 3, ..., M)$.

$$F_{fs} = [f_{fs}(\mu_1), f_{fs}(\mu_2), \ldots, f_{fs}(\mu_n)] \quad (4)$$

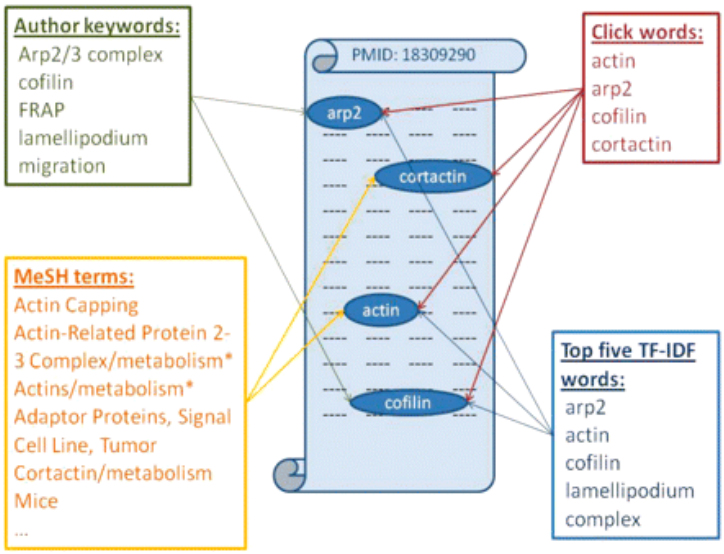

Figure 1: An example of click-words, top-scoring

Comparing click-words with other document keywords we found that, although there was overlap, user click-words were quite different from other types of important keywords (see Figure 3.1 for an illustration). Document keywords are all meant to capture the important contributions of a document, but they rely on different weighting mechanisms, which may be the reason for their difference.[13] Click-words are the product of click-through logs and they represent the 'wisdom of the crowds' as to what terms in an article may be important from the users' perspective. Top weighted TF–IDF words capture the importance of words with respect to other articles in a collection. In contrast, Pub Med relies on indexers to assign the appropriate Mesh indexing terms to Pub Med articles. As a result, these words are not immediately available for new articles. Moreover, they are not necessarily found in the title and abstract of the article. Author keywords, on the other hand, are not included in the MEDLINE citation. In addition, they are not easily procured—we found that they are available for only 13% of the articles in the Pub Med Central full text database.[12]

Each entry Wa,b of the adjacency matrix contains value of Wa,b that represents the degree of connectivity between the two pages *a* and *b*.

D. Harvest

Traditional information retrieval techniques use inverted lists to efficiently identify documents that contain the keywords in the query. In the same spirit, DBXplorer maintains a symbol table, which identifies columns in database tables that contain the keywords. Assuming index is available on the column, then given the keyword,

to efficiently find the rows that contain the keyword. If index is not available on a column, then the symbol table needs to map keywords to rows in the database tables directly.

Figure 2: relevant result processing

In the image Figure 2, the relevant result progressing is enhanced using the similarity matrix. The matrix is calculated by the following formula

$$T(pi, pj) = M(pi, pj) + S(pi, pj) \tag{5}$$

The semantic similarity is represented in terms of a semantic similarity matrix that gives the similarity score between every pair of Web pages. Thus, the semantic similarity matrix S is combined with the adjacency matrix M in order to derive the semantically enriched weight matrix T.

Table 1
Statistics of Experimental Data Set

Attributes	*Data Set-I*	*Data Set-II*	*Data Set-III*
Total Access Entries	9367	4575	1253
Total Web page accessed in log	85625	56588	52335
Total pages Identified by Crawler	5263	4521	1252
Different access users	850	1919	835
Total Identified sessions	9367	4575	1253

The above table gives the different statics of experimental data sets of the different attributes, from that Figure 3.2 shows the distribution of session length for the three data sets. For example, session length of two indicates the percentage of sessions with two page requests that occur in the collection of sessions. As shown in Figure 3.2, the session distribution of data sets are identified in the percentage of total sessions decreases when session length increases.

4. CONCLUSION

This paper we introduced the novel problem of predicting the effectiveness of keyword queries over DBs and showed that the current prediction methods for queries over unstructured data sources that cannot be effectively used to solve this problem. So we set forth principled framework and proposed novel algorithms to measure the

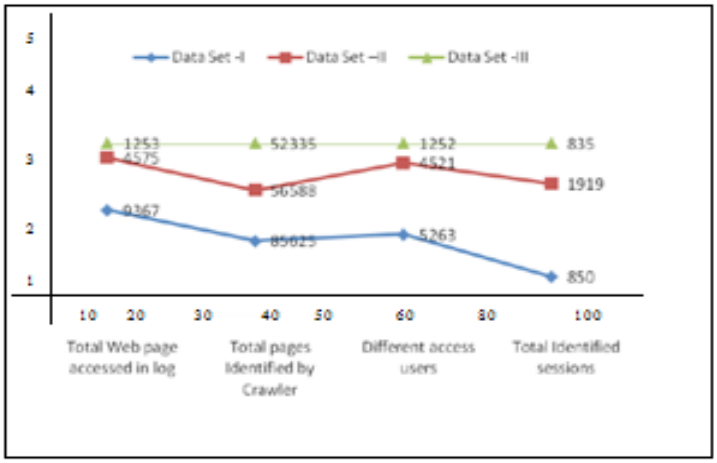

Figure 2: Distribution of session length

degree of the difficulty of a query over a DB, using the ranking robustness principle. Experimental results shows that the simple method performs better compare to existing information retrieval methods.

Appendix A

We used normalized pattern to evaluate the predict ranking, This is calculated in the rank list R with a log discount factor.

$$\mathrm{R} = \frac{1}{\mathrm{Z}} \sum_{\mathrm{K}} \frac{2r(\mathrm{K}) - 1}{\log(1+k)} \tag{6}$$

where, $r(k)$ denotes the target label for the kth ranked item in R and r is chosen for perfect ranking, we used top(n) for prediction ranking and these scores are averaged for all ranking list of comparison.

REFERENCES

[1] Cheng.S, Termehchy .A, and Hristidis .V.(2012). "predicting the effectiveness of keyword Queries on databases," in proc. 21st ACM int. CIKM, maui, HI, pp. 1213-1222.

[2] Collins-thompson .K and Bennett .P. N., (2010)"predicting query performance via Classification," in proc. 32nd ecir, milton keynes, U.k., pp. 140–152.

[3] Ganti.V, He.Y., and Xin.,(2010) "Keyword++: A framework to improve keyword search over Entity databases," in Proc. VLDB Endowment, Singapore, Sept., Vol. 3, No. 1–2, pp. 711–722

[4] Kurland.O., Shtok.A, Carmel.D., and Hummel.S., (2011)"A Unified framework for Post-Retrieval query-performance prediction," in Proc. 3rd Int. ICTIR, Bertinoro, Italy, pp15–26.

[5] Kurland.O, Shtok.A, Hummel.S, Raiber.F, Carmel.D, and Rom.O, (2012) "Back to the Roots: A probabilistic framework for query performance prediction," in Proc. 21st Int. CIKM, Maui, HI, USA, pp. 823–832.

[6] Shtok.A, Kurland.O, and Carmel.D, (2009)"Predicting query performance by query-drift estimation," in Proc. 2nd ICTIR, Heidelberg, Germany, pp. 305–312.

[7] Shiwen Cheng, Arash Termehchy, and Vagelis Hristidis (2014)"Efficient Prediction Of Difficult Keyword Queries Over Databases" IEEE Transactions on knowledge and data engineering, Vol. 26.

[8] Trotman and Wang.Q, (2010) "Overview of the INEX 2010 data centric track," in Vugh, the Netherlands, pp. 1–32, 9th Int. Workshop INEX 2010.

[9] Termehchy.A and Winslett.M.(2011) Using Structural Information in XML Keyword Search E_ectively. TODS, 36(1):4:1-4:39.

[10] Termehchy. A, Winslett.M, and Chodpathumwan.. Y. (2011) How Schema Independent Are Schema Free Query Interfaces? In ICDE, pages 649-660, 2011.

[11] Townsend.S.C, Zhou.Y(2002) and Croft. Predicting Query Performance. In SIGIR, pages 299-306.

[12] Shen D, et. al., AAAI'07: Proceedings of the 22nd National Conference on Artificial Intelligence. AAAI Press; 2007. Mining web query hierarchies from clickthrough data; p. 341-346

[13] Liu F, et. al., NAACL'09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics; 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts; p. 620-628.