Efficient Data Retrieval System for Text Search Queries

UmaMaheswari.C[1], M.N.Sushmitha[2]

School of Computing Sciences, Hindustan University
maheshuma49@gmail.com,mnsushmitha@hindustanuniv.ac.in

Abstract: ***The text filtering server monitors a stream of incoming documents for a set of users, who register their interests in the form of continuous text search queries. The objective of the server is to constantly maintain for each query a ranked result list providing the recent documents with the highest similarity to the query.I propose the solution for processing continuous text queries efficiently.The solution indexes the streamed documents in main memory with a structure based on the principles of the inverted file, and processes document arrival and expiration events with an incremental threshold-based method. Based on incremental threshold algorithm, the order of the links will be displayed.***

Keywords: Continuous queries,Document streams,Text filtering

1. Introduction

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Considerable research effort has been invested in improving the effectiveness of information retrieval systems. Techniques such as relevance feedback, thesaural expansion, and pivoting all provide better quality responses to queries when tested in standard evaluation frameworks. But such enhancements can add to the cost of evaluating queries. In this paper we consider the pragmatic issue of how to improve the cost-effectiveness of searching. It describes a new inverted file structure using quantized weights that provides superior retrieval effectiveness compared to conventional inverted file structures when early termination heuristics are employed. That it is able to reach similar effectiveness levels with less computational cost, and so provide a better cost/performance compromise than previous inverted file organizations[6].

It extend the applicability of impact transformation, which is a technique for adjusting the term weights assigned to documents so as to boost the effectiveness of retrieval when short queries are applied to large document collections. In conjunction with techniques called quantization and thresholding, impact transformation allows improved query execution rates compared to traditional vector-space similarity computations, as the number of arithmetic operations can be reduced. The transformation also facilitates a new dynamic query pruning heuristic. It give results based upon the *TREC* web data that show the combination of these various techniques to yield highly competitive retrieval, in terms of both effectiveness and efficiency, for both short and long queries [7].

Information filtering systems based on statistical retrieval models usually compute a numeric score indicating how well each document matches each profile. Documents with scores above profile-specific *dissemination thresholds* are delivered. An optimal dissemination threshold is one that maximizes a given utility function based on the distributions of the scores of relevant and non-relevant documents. The parameters of the distribution can be estimated using relevance information, but relevance information obtained while filtering is *biased*. This paper presents a new method of adjusting dissemination thresholds that explicitly models and compensates for this bias. The new algorithm, which is based on the Maximum Likelihood principle, jointly estimates the parameters of the density distributions for relevant and nonrelevant documents and the ratio of the relevant document in the corpus. Experiments with TREC-8 and TREC-9 Filtering Track data demonstrate the effectiveness of the algorithm[3].

Ranking techniques are effective attending answers in document collections but can be expensive to evaluate. It propose an evaluation technique that uses early recognition of which documents are likely to be highly ranked to reduce costs; for our test data, queries are evaluated in 2% of the memory of the standard implementation without degradation in retrieval effectiveness. CPU time and disk tra_c can also be dramatically reduced by designing inverted indexes explicitly to support the technique. The principle of the index design is that inverted lists are sorted by decreasing within-document frequency rather than by document number, and this method experimentally reduces CPU time and disk tra_c to around one third of the original requirement. We also show that frequency sorting can lead to a net reduction in index size, regardless of whether the index is compressed[5].

2. Existing System

The currently used search engine does not provide efficient results for the given query.It provides only the older information which is already residing in the database.In the results for the given query,it will display the repeated documents or links.It doesn't maintain a ranked result list for each query constantly for continuous text search queries.It doesn't provide efficient information constantly.Based on

the time boxed criteria,datas are not provided.Hence updated information is not given to the user.

3. Proposed System

In this proposed system, it proposes the first solution for processing continuous text queries efficiently. Our objective is to support a large number of user queries while sustaining high document arrival rates.In this Project,Proposes the scheme of Sliding window which updates the latest information with respect to time and updated data. We also propose to remove the duplication of the Records in the web content. It also get the feedback from the previous users of the corresponding website. So finally it implements the ranking process by getting the feedback from the users, removal of duplication along with sliding window approach.

A Proposed System Architecture

Figure 1 Architectural design is the high level design where the whole system is divided into different subsystems and the dependency relationship and communication between them are also identified.A good architectural design shows the dependencies and the primary communication mechanisms between the various packages.

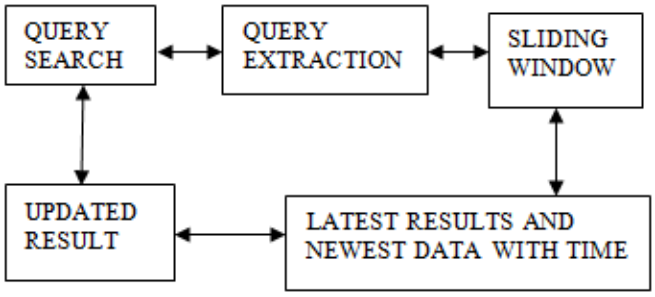

Figure 1: Proposed System Architecture

4. Algorithms

Stemming Algorithm

Stemming is the process for reducing inflected or sometimes derived words to their stem, base or root form generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since 1968. Many search engines treat words with the same stem as synonyms as a kind of query broadening, a process called conflation. Stemming programs are commonly referred to as stemming algorithms or stemmers. Stemming is the process of grouping different inflected forms of a word to determine the lemma for that word. A lemma is the base form of the word and may change when inflected, while a stem does not change. For example, for the inflected word ‘produced’, the lemma is ‘produce’ while the stem is ‘produce’ as there is an inflected form like ‘production’. As a result, stems are not necessarily complete words.

Unsupervised Learning Algorithm

Unsupervised learning refers to the problem of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning. Unsupervised learning is closely related to the problem of density estimation in statistics . However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods.

Incremental Threshold Algorithm

Threshold Algorithm is to employ threshold-based techniques to derive the initial result for a query, then continue to update the thresholds to reflect document arrivals and expirations. The thresholds are used to incrementally maintain the query result, and also to avoid processing documents that have too low a similarity score to affect the result.

5. Modules

Query Search

The user searches for the information, currently for the ordinary search of the information lot of results will be displayed with the duplications and the random display of outputs. In our project the user is going to give the query and accordingly the Updated information is provided.

Query Extraction

Consider a search in an internet for a topic; it will display the details with all possible output with repetitive information and the updates at the last in the displayed order. The data server used in this module is used in order to retrieve the exact updated information with no duplicates by the use of sliding window technique and with the help of Unsupervised Duplication Detection (UDD). Hence data server performs the above operation and gives the result for the given query.

Sliding Window

The sliding window is the technique which utilizes both time and data updations. In case of time, if the information provided in the site is updated later then the last updated data
will be displayed first. In case of data, if the old data is updated then the updated information will be displayed according to the priority.

Duplication Detection

In this project, to avoid the repetition of same data result we utilize Unsupervised Duplication Detection algorithm. By using this algorithm, we can avoid the duplicate data and hence as a result updated and without duplicate result will be displayed to the user.

Result Rating

After achieving the refined search results, the user will be asked to provide the feedback regarding the searched site. Based upon the feedback, ranking will be provided for the site. If the user provides the like feedback then the ranking will be incremented and if the feedback is unlike, then the ranking will be decremented. Accordingly the best ranking results will be displayed in that refined search site.

Updated Result

Thus doing the search by means of these techniques, data retrieval will be efficiently done. The advancement in it is the data duplication is avoided and also the latest innovated information is provided to the user.

6. Conclusion

Efficient updated data retrieval system is used for processing of continuous text queries over document stream. These queries have a set of search terms, where continuous monitoring is done and result is provided in updated manner.Two incremental threshold techniques such as EIT and LIT are useful in providing the updated results.

7. Future Works

Future works are one of the inevitable for any kind of software project.some of the enhancement features that are applicable for our project is listed as follows:In future we will extend our documents tagged with metadata and apply special scoring mechanisms in this type of documents.one more technique is to extract updated datas from the database itself.

References

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and Issues in Data Stream Systems,” Proc. Twenty-First ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS ’02), pp. 1-16, 2002.
[2] J. Zobel and A. Moffat, “Inverted Files for Text Search Engines,” ACM Computing Surveys, vol. 38, no. 2, pp. 1-55, July 2006.
[3] Y. Zhang and J. Callan, “Maximum Likelihood Estimation for Filtering Thresholds,” Proc. 24th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’01), pp. 294-302, 2001.
[4] K. Mouratidis, S. Bakiras, and D. Papadias, “Continuous Monitoring of Top-k Queries over Sliding Windows,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’06), pp. 635- 646, 2006.
[5] M. Persin, J. Zobel, and R. Sacks-Davis, “Filtered Document Retrieval with Frequency-Sorted Indexes,” J. Am. Soc. for Information Science, vol. 47, no. 10, pp. 749-764, 1996.
[6] V.N. Anh, O. de Kretser, and A. Moffat, “Vector-Space Ranking with Effective Early Termination,” Proc. 24th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’02), pp. 35-42, 2001.
[7] V.N. Anh and A. Moffat, “Impact Transformation: Effective and Efficient Web Retrieval,” Proc. 25th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’02), pp. 3-10, 2002.
[8] H.R. Turtle and J. Flood, “Query Evaluation: Strategies and Optimizations,” Information Processing Management, vol. 31, no. 6, pp. 831-850, 1995.
[9] M. Kaszkiel, J. Zobel, and R. Sacks-Davis, “Efficient Passage Ranking for Document Databases,” ACM Trans. Information Systems, vol. 17, no. 4, pp. 406-439, 1999.
[10] T. Strohman, H. Turtle, and W.B. Croft, “Optimization Strategies for Complex Queries,” Proc. Research and Development in Information Retrieval (SIGIR ’05), pp. 219-225, 2005.

Authors Profile

UmaMaheswari.C, is a final year student of M.Tech(CSE) in Hindustan University, Chennai, India.

M.N.Sushmitha, M.Tech (CSE), Assistant Professor, Experience: 7 years, Selection Guide, Hindustan University, Chennai, India.