Extremely Pertinent Document Assortment from Disseminated Databases

Ahmad Alkhodre, Turki Alghamdi, Mohammad Husain
Department of Computer Science & IS
Islamic University, Madina
Saudi Arabia
aalkhodre@gmail.com, dr.turki.iu@gmail.com, mohd.husain90@gmail.com

***ABSTRACT*: *With the rapid growth of computer technology in the recent years, electronic information has been rapidly increased and become more and more important to help users access a huge volume of web pages. This is really noticeable to find the information in efficient and effective manner especially on the web, when we are working in heterogeneous environment. As the availability of information increases on the web, the need of finding proper and accurate information is also a challenging task. There are the numerous ways to find the information on the web but the ultimate need nowadays is that of predicting the user needs in order to improve the usability and user retention. In this paper we have proposed an algorithm which shows superior performance compared to other existing methods and that will provide most accurate, fast and relevant information in front of the user.*

Keywords: Personalization, Meta-search, Heterogeneous, Feedback, Indexing

Received: 10 November 2014, Revised 18 December 2014, Accepted 23 December 2014

1. Introduction

To provide users with only relevant data from the huge amount of available information, personalization systems resort to user preferences to allow users to express their interest on specific parts of data. When there is huge amount of document exist and the query fired to get the best result from existing document then the process is termed as information retrieval system. The documents may contain the same type of information or may have different type of information among them.

Information filtering systems process a document stream and recommend relevant documents to individual users.

Many retrieval systems nowadays reference documents from various sources or collections. These sources may reside on the same computer as the retrieval system or may be dispersed over different locations. Merging is basically done when we have To provide users with only relevant data from the huge amount of available information, personalization systems resort to user two or more sorted files and combining them into one file. When large data sets are required to merge together, it adds significant complications to data manipulation process. Therefore, performing efficient merging becomes very critical, especially with an implementation with large datasets. Merging data sets is to combine two or more data sets horizontally by matching the keys from both data sets[1].

The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would look into every document in the corpus, which would require considerable time and computing power. The effectiveness of information retrieval systems is generally expressed in terms of relevant and non-relevant documents being retrieved or not retrieved. Most retrieval performance measures are based to some extent on the ordering of documents of varying degrees of expected relevance. Relevant documents may be loosely described as those documents that address the user's needs[1].

There are a enormous amount of strategies and techniques available among them a meta-search engine can use in its goal to find the required documents. Associated with these various ways, though, can be vastly different levels of performance which can be demonstrated by various normalized measures such as precision, recall, mean reciprocal rank [5], etc. The key challenge for a meta-search engine is to take an information need, expressed as a query, submit it to the other search engines, and attempt to obtain the same results that would be generated if the documents accessible via those other search engines were all in a single collection. This is the essence of the distributed information retrieval problem.

However, the inaccurate and poor qualities queries have been reported as one of the main reasons of unsatisfying performance of search engines [6]. Generally, users cannot exactly provide the requirement of their information needs, and even the well-formulated queries may lead to few hits, because users have limited knowledge of the exact documents and may type in mismatched keywords. The uncommon, nonspecific and sometimes ambiguous queries always lead to non-relevant and even puzzling results.

Interactive query expansion is a technique used to improve this problem by suggesting persistence to a typed query, and thus help the user to formulate a better query. This may significantly increase the efficiency of the retrieval process. When implementing interactive query expansion the users are not able to choose expansion terms better than a system could in automatic query expansion.

The study recommended that how the user is presented with interactive query expansion is highly important for its effectiveness (Figure 1). Real time query expansion is a form of interactive query expansion that integrates the query expansion into the process of query formulation[4].

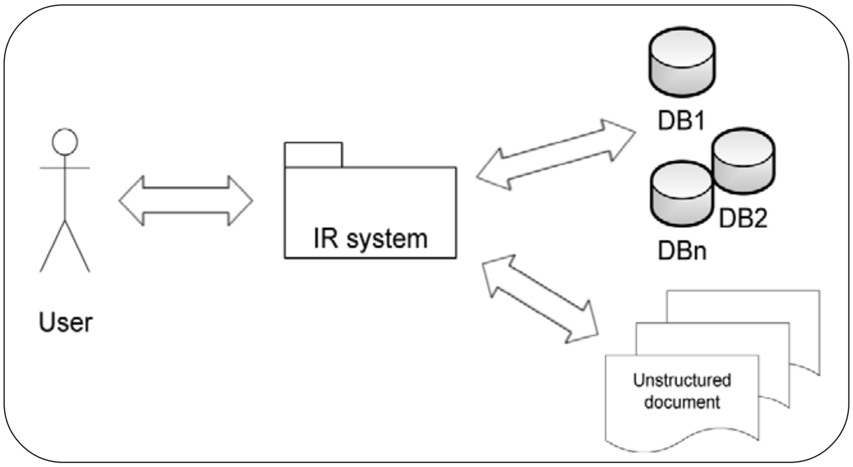

Figure 1. Information Retrieval System

2. Literature Review

Robert M. Losee and Lewis Church Jr.[1] presented a methodology for finding the k best unique documents to be presented to the end user, since many searches on the Internet are high precision searches. Generally, when retrieving documents from each separate database, it may be desirable to retrieve the best b documents from each database, rather than retrieving all the documents from each database. The major emphasis of this paper is on analytical techniques for predicting the performance of various collection fusion scenarios.

This paper makes the following contributions in predicting system retrieval performance:

1. methods for estimating the precision of the system and 2. methods for predicting retrieval performance measured as Average Search Length.

Luo Si and Jamie Callan[3] have suggested that the CORI algorithm has been one of the most consistently effective resource ranking algorithms since it was introduced. This paper identifies a previously unknown weakness that is triggered in environments containing many small databases and a few very large databases, as might occur in corporate or government environments.

The proposed algorithm uses constants to model the probability or relevance given a document, and to determine how much of a centralized complete database ranking to model. Experimental results indicate that the algorithm is effective across a range of parameter settings, but these constants are nonetheless ad-hoc and can be considered a weakness of the algorithm.

Byoung-Tak Zhang and Young-Woo Seo[7] have suggested that the retrieved documents undergo preprocessing, here we have used standard term-indexing techniques, such as removing stop-words and stemming. This is done by learning the profiles of users. A user profile consists of one or more topics. Topics represent user's information needs. It always needed to update by adding new terms, removing existing terms, and modifying term weights. The first experiment was to compare the performance of the proposed method with the conventional feedback methods.

Yves Rasolofo et. al.[8] investigated the problem of results merging to predict the relevance of collections to a given query, analyze a limited number of full documents (e.g., the top five documents) retrieved from each collection and then consider term proximity within them.

In this approach, the broker broadcasts the query to all available collections (|C| collections), with each collection returning nb_doc highly ranked documents to the broker. The broker then calculates the score for each document received (nb_doc * |C|), and sorts them according to their scores, with the collections matching the n_first documents being selected.

a) A combination of our two strategies works better than other collection-selection/results merging combinations,

b) our selection method works well even with RSM or CORI merging strategies, and our merging approach works also well when no selection is performed.

however, our selection strategy requires more transfer traffic for the downloading of the first nb_doc (nb_doc=5 in our evaluations) documents per collection. Thus, response delay may increase slightly.

Keith L. Clark and Vasilios S. Lazarou [9] have presented Sites managed as local deductive databases and WWW as a distributed deductive one. Prolog-like query language provides expressiveness and accuracy concerning the user needs. The predicate set for query formulation corresponds to characteristic document properties. Completely precise answers. The semantics behind the used terms is captured; polysemy and synonymy are tackled. Fully distributed, scalable and modular system; the information providers are not passive request servers.

Fidel Cacheda, VassilisPlachouras, IadhOunis [10] have established the bottlenecks and limitations of each possible configuration. Our work extends previous works with respect to both the size of the collection and the number of query servers simulated, and confirms some of the previous findings in the literature. Indeed, we have identified two main bottlenecks in a distributed and replicated IR system: the brokers and the network. The load on the brokers is mainly due to the number of local answer sets to be sorted(characteristic of a distributed system), as it was also shown by Cahoon and McKinley (1996). Therefore, the load can be improved by reducing the number of documents included in the local answer sets by all the query servers, which could affect precision and recall. Another way is to reduce the number of local lists sent to the brokers, by designing more complex and elaborate distributed protocols.

Lanbo Zhang, Yi Zhang et. al.[11] have envisions two faceted feedback solicitation mechanisms, and propose a novel user profile-learning algorithm that can incorporate user feedback on features. To evaluate the proposed work, we use two data sets from the TREC filtering track, and conduct a user study on Amazon Mechanical Turk. Our experimental results show that user feedback on faceted features is useful for filtering. The new user profile-learning algorithm can effectively learn from user feedback on faceted features and performs better than several other methods adapted from the feature-based feedback techniques proposed for retrieval and text classification tasks in previous work.

3. Proposed Methodology and Architecture

3.1 Architecture

In this we retrieve the data from different sources after that applied the technique to merge them, when merging takes place then perform indexing which provides preprocessing source from which we extract documents and ready to interact with IR system.

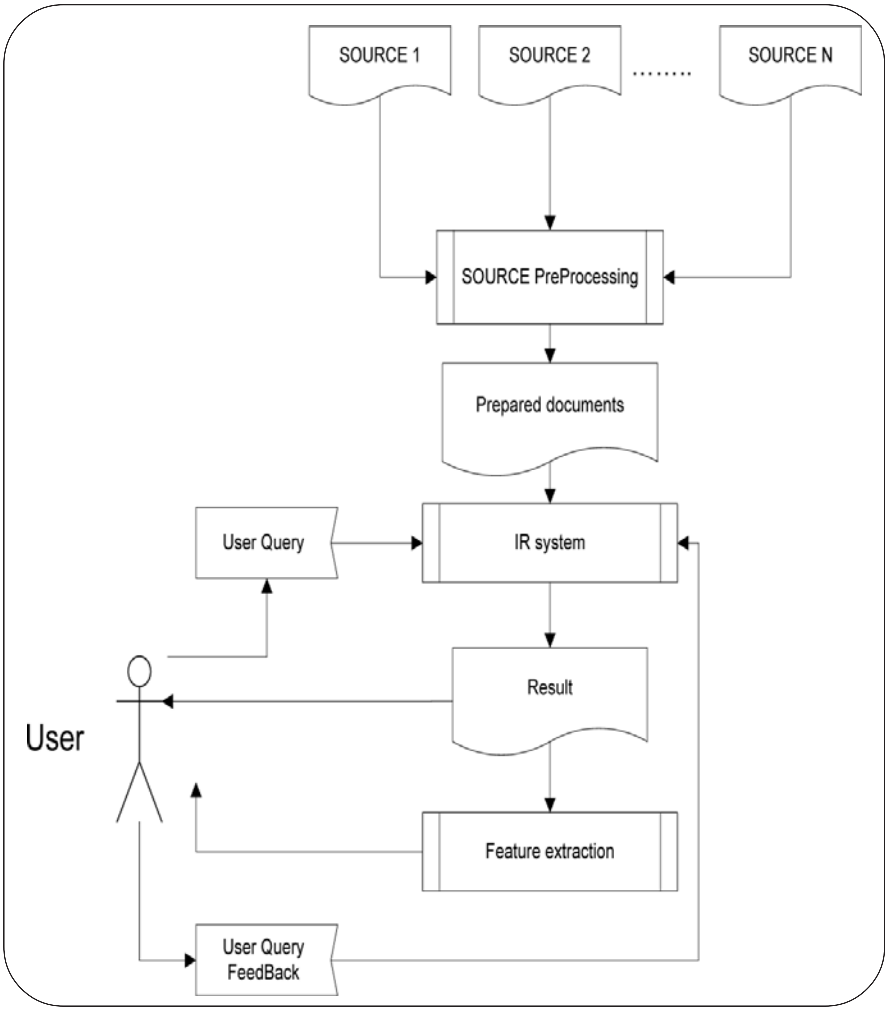

Figure 2. Architecture for retrieving the pertinent document from heterogeneous databases

Now user poses the query and get the result from information retrieval system, the result is provided to the user for further extraction purposes, if it is required then user provides the feedback and again query interact with the IR system based on user query feedback and finally it gives the extracted result to the end user.

3.2 Proposed Algorithm

We want to select those documents from number of sources, which satisfy our query ‘q’. The Basic idea of this algorithm is that we examine different type of data sources in the order Source1, Source2, Source3, Source4, Source5,………., SourceN, until we get the Sources which contain the query ‘q’. This algorithm works as follows:

1. Search preparation

a. Index the sources document selected

b. Merging the heterogonous Databases 2. Input the user query “q1”

3. Examine each source with its documents accumulated in it. If any document of source contains the query at least one time then we select that Data Source.

4. If not all the documents of source contain the query then that database will not be selected.

5. We select only those documents from each source in which the query ‘q1’ occurs at least one time.

6. Extract features from the resulted document in step 5

7. Display the result to the user

8. Input the user feedback “q2”

9. Result filtering

    a. Document classification based on “qory2”

    b. Remove the document that is not satisfied user feedback

    c. Rank all the classified documents according to the no. of occurrence of query ‘q2’ in descending order.

10. Return the top ‘n’ most relevant documents from the ranked list of documents for any positive integer ‘n’.

4. Conclusion

To find relevant document from heterogeneous databases in efficient and effective manner is the major issue now a days. There are number of techniques available to solve this issue. Based on our study we found that even there are some vacuum exist. To bridge the gap we have proposed an architecture and a novel algorithm for merging the heterogeneous databases which contains the given query and from those databases we select top most relevant document, secondly we have also introduced users feedback technique which provides appropriate document by interacting again with the information retrieval system. Our proposed method can yield substantial improvements over existing techniques.

References

[1] Robert M., Losee., Lewis Church Jr. (2003). Information Retrieval with Distributed Databases: Analytic Models of Performance, *IEEE Transactions on Parallel and Distributed Systems*, 14 (12), December.

[2] Deepika Sharma., KirtiChoudhary., Sandeep Kumar Poonia. (2014). Design And Implementation Of Context Based Information Retrieval System, *International Journal of Engineering, Management & Sciences* (IJEMS), 1 (6), June.

[3] Luo Si., Jamie Callan. (2003). Relevant Document Distribution Estimation Method for Resource Selection, SIGIR ’03, July 28-Aug 1, Toronto, Canada.

[4] Sigurd Wien., Herindrasana Ramampiaro. (2013). Efficient Top-K Fuzzy Interactive Query Expansion While Formulating a Query, *Norwegian University of Science and Technology*, 2013.

[5] Voorhees, E. M., D. K. H., eds. (2000). The TREC-8 Question Answering Track Report, Proc. Eighth Text REtrieval Conf. (TREC-8), *Nat’l Inst. of Standards and Technology*.

[6] Ju Fan., Hao Wu., Guoliang Li., Lizhu Zhou. (2010). Suggesting Topic-Based Query Terms as You Type, 2010 12th International Asia-Pacific Web Conference.

[7] Byoung-Tak Zhang., Young-Woo Seo. (2001). Personalized Web-Document Filtering Using Reinforcement Learning, *journal of Applied Artificial Intelligence*, 15 p. 665-385.

[8] Yves Rasolofo et. al. (2001). Approaches to Collection Selection and Results Merging, *In*: Proceedings of the tenth international conference on *Information and knowledge management*, p. 191-198.

[9] Keith, L., Clark., Vasilios, S., Lazarou. (1997). A Multi-Agent System for Distributed Information Retrieval on the World WideWeb, Enabling Technologies: Infrastructure for Collaborative Enterprises, *In*: Proceedings., Sixth IEEE Workshops.

[10] Fidel Cacheda., VassilisPlachouras., IadhOunis. (2005). A case study of distributed information retrieval architectures to index one terabyte of text, *Information Processing & Management*, 41 (5), September, p. 1141–1161

[11] Lanbo Zhang., Yi Zhang., et. al. (2011). Filtering Semi-Structured Documents Based on Faceted Feedback, SIGIR ’11 *In*: Proceedings of the 34th international ACM SIGIR conference on *Research and development in Information Retrieval* p. 645-654, 2011.