ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 392 - 394 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Answering General Time Sensitive Queries

[1]O.Ramesh, [2]V. Vidya Sagar
[1]Final M.Tech Student, [2]Assistant Professor
[1,2]Dept of Computer Science and Engineering
[1,2]Pydah College of Engineering, Visakhapatnam, AP, India.

Abstract: Clustering is one of the knowledge extraction techniques in data mining, now a days time relevance of retrieval query results takes more importance in search engines, because user gives importance to latest and relevant information. In this paper we are proposing an efficient technique clustering technique based on time sensitivity of the documents. Our experimental results show more accurate and efficient results than traditional approaches.

INTRODUCTION

Information retrieval is the process of getting information links relevant to an information need from a collection of information resources. Searches can be based on metadata or on full-text (or other content-based) indexing. It is automated computerized information retrieval systems are used to decrease what has been called information overwhelm. Many public libraries use IR systems to provide access to books and journals and other documents. Web search engines are the most IR applications.

An information retrieval process starts when a user gives a query into the system. Queries are formal formatted statements of information require and consider an example search texts in web search engines. In information retrieval a query does not distinctly find a single object in the collection. Instead of this more items may match the query with various degrees of relevancy.

An object is an entity that is represented by information in a database. User questions are matched against the database information. Depends upon the application the data objects may be such as text documents/images/audio/mind maps or videos. The documents are not kept or stored directly in the IR system but are instead notated in the system by document or metadata. Many of the IR systems calculate a numeric score value on how well every object in the database matches the query and rank the data objects considering to this value. The most top ranking data objects are then shown to the user.

The method may then be iterated if the user wishes to modify the query. For fast retrieving of correct documents by IR strategies the documents are converted into a user understandable representation. Each retrieval scheme incorporates a particular model for its document representation purposes. The models are categorized according to two types such as the mathematical basis and the properties of the model.

There are three types of similarity measure: association, correlation and distance. Clustering is the process of grouping items in a way that items in a group are similar to each other and dissimilar to the items in other groups. In information retrieval is the problem for getting the desired documents given a short formatted query in the confusion of the query. Tokens in a query can occur in different documents and contents usually it is more difficult to find the documents where a search token is used in the intended sense. So many search results in a result list of a classic information retrieval system are not difficult. Clustering results for information retrieval is a way to this problem by structuring the large list of search results. This will not reduce any mismatched documents but if the cluster labels are intuitive enough the user can self-choose the cluster which consists of the more relevant documents for his request.

Informational retrieval can be done based on exact match of keywords or candidate set generations can be done by string transformation, localization and globalization techniques improves the search process of search engine optimization. To fulfill user search goals, search results can be mined based on clustering techniques of pattern mining approaches.

Relevant information can be clustered based on the distance or similarity between the data objects, distance can be considered for numerical data and similarity can be considered for categorical information or objects. Basic clustering mechanism like k means algorithm works with random selection of the centroid and these results depends on selection of the centroid and this approach not applicable at density based clustering approaches.

RELATED WORK

Sequential scanning of the text: extra memory is in the worst case a function of the query size, and not of the database size. On the other hand, the running time is at least proportional to the size of the text, for example, string searching.

Indexed text: An "index" of the text is available, and can be used to speed up the search. The index size is usually proportional to the database size, and the search time is sub-linear on the size of the text, for example, inverted files and signature files. Formally, we can describe a generic searching problem as follows: Given a string t (the text), a regular expression q (the query), and information (optionally) obtained by preprocessing the pattern and/or the text, the problem consists of finding whether $t\epsilon\sum*q\epsilon\sum*$ (q for short) and obtaining some or all of the following information:

1. The location where an occurrence (or specifically the first, the longest, etc.) of q exists. Formally, if $t\epsilon\sum*q\epsilon\sum*$, find a position m >=0 such that $t\epsilon\sum$(from 0 to m)$q\epsilon\sum*$. For example, the first occurrence is defined as the least m that fulfills this condition.
2. The number of occurrences of the pattern in the text. Technically it is the number of all possible values of m in the previous category.
3. All the locations where the pattern occurs (the set of all possible values of m). In general, the complexities of these problems are different.

The efficiency of retrieval algorithms is very important, because we expect them to solve on-line queries with a short answer time. This need has triggered the implementation of retrieval algorithms in many different ways: by hardware, by parallel machines, and so on.

This class of algorithms is such that the text is the input and a processed or filtered version of the text is the output. This is a typical transformation in IR, for example to reduce the size of a text, and/or standardize it to simplify searching.

The most common filtering/processing operations are:

- Common words removed using a list of stop words;
- Uppercase letters transformed to lowercase letters;
- Special symbols removed and sequences of multiple spaces reduced to one space;
- Numbers and dates transformed to a standard format;
- Word stemming (removing suffixes and/or prefixes);
- Automatic keyword extraction;
- Word ranking.

Searching user interesting results in search engines is still an important research issue in the field of knowledge and data engineering, Even though various approaches available for finding the results for user query they are m may not be optimal. In this paper we are proposing an efficient file relevance score mechanism followed by the clustering mechanism, In the clustering approach we cluster the retrieved documents based on the time relevance.

PROPOSED SYSTEM

In this paper we are proposing an integrated approach of user search results with file relevance score with the basic factors term frequency(TF) that indicates the number of occurrences of the document and (IDF) i.e. number of occurrences of the keyword with respect to the all the documents along with their time stamps and clustering. Initial phase involve file relevance score and second phase involves the clustering approach.

We proposed a novel file relevance score measurement with number of terms in the file, number of occurrences of the term (term frequency) and number of files.

```
relevance_Scores[j] = Convert.ToDecimal((1 /
termsinfile[j]) * (1 + Math.Log(termfreqs[j])) *
Math.Log(1 + (filecount / numberoffiles)));
```

Ranking function calculates the term frequency and inverse document frequency for finding the score of the query or keyword with respect to the files, and forwards the datasets according to the score to the user based on ranking.

Search Query

In this module the user enter the query for searching of related files in the web services. After entering of query the search engine will search query related files in the web storage and retrieve that file. The search engine will find the files based on the query string. The search engine will take string from the query string and comparison each and every file in web storage. If any related files are matched through query only that file can be retrieve and sent to users.

File Relevance

In this we are find out relevance of each query related file. The relevance of file can be calculated by the total number of occurrence of query string in file by total number of words in that file. In this module we are find out the individual of total number words in the file and occurrence of query string in that file. The relevance of file can calculated by query string matching files. The relevance of file can be specified by percentage of occurrence query string in file.

Clustering

In this module we are find out the newly query related file based on time factor of that file. Before finding newly query related file we are cluster the same category of file. The cauterization of can be done by using k_Medoids clustering algorithm. The procedure of K-Mediods algorithm as follows.

Step1: Randomly select centroid based on the number of cluster are given

Step2: Find the Euclidian distance of each centroid to other files and find out the nearest distance of file. Finding of nearest distance is based on time of file can be upload.

Step3: After finding nearest distance then cluster those documents.

Step4: In that clusters we are again randomly select centroid and find out distance of each file from the centroid based on the time of file can be uploading.

Step5: This step can repeat for finding nearest distance of each file.

Step6: After finding we can cluster those documents.

In clustering process initial, it receives the number of clusters as input parameter then randomly select the k number of centroid from the retrieved results, now calculates the Euclidian distance between the centroid and all the documents with respect to the time stamps, continue the process until a maximum number of user specified iterations or until no changes made in the cluster.

CONCLUSION

Now a day's search of queries in the web storage and also find retrieved documents are newly or not. By finding related documents is based the occurrence of words in that document, for purpose of finding related documents we are also concentrate on that documents are newly one or not. In this we are using clustering algorithm for clustering related document. In this paper we are using k_medoids algorithm clustering same type of documents and also retrieve the newly document. By proposing this approach we are performing the fast string searching and also retrieve the newly query string related documents. This approach is an efficient one for the clustering of newly and related query string documents.

REFERENCES

[1] J. Ponte and W. B. Croft, "A Language Modeling Approach to information retrieval". *Proceedings of the 21st annual international ACM SIGIR conference*, 275-281, 1998.

[2] F. Song and W. B. Croft. "A general language model for informationretrieva". *Proceedings of the 22nd annual international ACM SIGIR conference*, 279-280, 1999.

[3] D. Hiemstra. *Using language* models *for information retrieval*. PhD thesis, University of Twente, 2001.

[4] J. Lafferty and C. Zhai. "Document language models, query models, and risk minimization for information retrieval". *Proceedings of the 24th annual international ACM SIGIR conference*, 111-119, 2001.

[5] V. Lavrenko and W. B. Croft. "Relevance-based language models". *Proceedings of the 24th annual international ACM SIGIR conference*, 120-127, 2001.

[6] D. Miller, T. Leek, and R. Schwartz. "A Hidden Markov Model information retrieval system". *Proceedings of the 22nd annual international ACM SIGIR conference*, 214-221, 1999.

[7] Swan, R. and Allan, J. "Automatic Generation of Overview Timelines". *Proceedings of SIGIR 2000 Conference*, Athens, 49-56, 2000.

[8] Swan, R. and Jensen, D. "Time Mines: Constructing Timelines with Statistical Models of Word Usage". *Proceedings of KDD 2000 Conference*, 73-80, 2000.

[9] J. Allan, R. Gupta, and V. Khandelwal. "Temporal Summaries of News Topics". *Proceedings of ACM SIGIR 01 conference*, 10-18, 2001.

[10] J. Pustejovsky, "TERQAS: Time and Event Recognition for Question Answering Systems", ARDA Workshop, MITRE, Boston (2002). (http://www.cs.brandeis.edu/~jamesp/arda/time/index.html)

[11]K. Wessel, W. Thijs, and H. Djoerd. "The Importance of Prior Probabilities for Entry Page Search", roceddings of SIGIR 2002, 27-34.