ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Automatic Analysis Technique in Order to Deduce Search Results From Search Engines for Efficient Machine Processing

Kadambala Sankar
M.Tech (Software Engineering),
Department Of Computer Science and Engineering,
Sarada Institute of Science Technology and
Mangement, Srikakulam.

Chintada Sunil Kumar
Assistant Professor,
Department of Computer Science and Engineering,
Sarada Institute of Science Technology and
Mangement, Srikakulam.

Abstract:

The word "Search engine" consists of two words: Search means to find something and engine means the procedures that find the specified information. So it's meaning can be clearly understood from its name. i.e. a search engine is a utility that provides the uses to find any information on the World Wide Web within a few seconds. Essentially search engines provide easy access to large databases of information. The search is generally carried out based on the similarity of the documents being searched for. The results are presented as per ranking of the items. However, for best ranking mechanism only similarity is not adequate. For some class of queries time could be an important dimension for searching in addition to the content similarity. Such queries are known as time sensitive queries that are processed and ranked based on the publication time and similarity. The existing research focused on retrieving recent queries.

Recently Dakka et al. presented a general framework for handling time – sensitive queries. In this paper we propose a framework that extends their work by considering more time related dimensions such as republication date and time, review articles of the documents with later dates etc. This improves the robustness of the system with respect to answering time sensitive queries as it can make use of review articles and summarize events in temporal domain. Thus the system is made capable of analyzing contents of web documents on different dimensions in addition to their publication date.

We built a prototype application to demonstrate the proof of concept. The empirical results revealed that the proposed framework for multi-dimensional time sensitive queries is effective. Our survey shows that the projected method is extremely required in the current scenario of Internet shopping boom in India. Our experiments indicate that the proposed approach is highly effective.

Keywords:

web database, Data interpretation, Search, dynamic search, Data arrangement.IntroductionInternet search engines are special sites on the Web that are designed to help people find information stored on other sites. There are differences in the ways various search engines work, but they all perform three basic tasks:

1.They search the Internet -- or select pieces of the Internet -- based on important words.

2.They keep an index of the words they find, and where they find them.

3.They allow users to look for words or combinations of words found in that index.

Early search engines held an index of a few hundred thousand pages and documents, and received maybe one or two thousand inquiries each day. Today, a top search engine will index hundreds of millions of pages, and respond to tens of millions of queries per day.

A web database is a system for storing information that can then be accessed via a website. For example, an online community may have a database that stores the username, password, and other details of all its members. The most commonly used database system for the internet is MySQL due to its integration with PHP — one of the most widely used server side programming languages. At its most simple level, a web database is a set of one or more tables that contain data. Each table has different fields for storing information of various types. These tables can then be linked together in order to manipulate data in useful or interesting ways. In many cases, a table will use a primary key, which must be unique for each entry and allows for unambiguous selection of data.

A large portion of the deep web is database based, i.e., for many search engines, data encoded in the returned result pages come from the underlying structured databases. Such type of search engines is often referred as Web databases (WDB). A typical result page returned from a WDBhas multiple search result records (SRRs). Each SRR contains multiple data units each of which describes one aspect of a real-world entity. Fig.shows three SRRs on a result page from a book WDB. Each SRR represents one book with several data units, e.g., the first book record in Fig. 1 has data units "Talking Back to the Machine: Computers and Human Aspiration," "Peter J. Denning," etc. In this paper, a data unit is a piece of text that semantically represents one concept of an entity. It corresponds to the value of a record under an attribute. It is different from a text node which refers to a sequence of text surrounded by a pair of HTML tags. Section describes the relationships between text nodes and data units in detail. In this paper, we perform data unit level annotation.

There is a high demand for collecting data of interest from multiple WDBs. For example, once a book comparison shopping system collects multiple result records from different book sites, it needs to determine whether any two SRRs refer to the same book. The ISBNs can be compared to achieve this. If ISBNs are not available, their titles and authors could be compared. The system also needs to list the prices offered by each site. Thus, the system needs to know the semantic of each data unit. Unfortunately, the semantic labels of data units are often not provided in result pages. For instance, in Fig. 1, no semantic labels for the values of title, author, publisher, etc., are given.

Having semantic labels for data units is not only important for the above record linkage task, but also for storing collected SRRs into a database table (e.g., Deep web crawlers) for later analysis. Early applications require tremendous human efforts to annotate data units manually, which severely limit their scalability.

Existing System:

Beyond asking for explicit user input, earlier work by focused on handling recency queries, which are queries that are after recent events or breaking news. Examples of recency queries are [NYC crane collapse] in May 2008, or [Sarkozy French elections] in April 2007. Li and Croft's time-sensitive approach processes a recency query by computing traditional topic-similarity scores for each document, and then "boosts" the scores of the most recent documents, to privilege recent articles over older ones. After retrieving documents give ranking of each documents based on number of words to be matched. Which document contain highest number of matching words those documents we can give highest rank. For suppose the highest ranking document is older one so that is not a recent one.In this situation your face a problem of not getting recent document. This can be overcome by proposing the given concept.

Disadvantages of Existing System:

If ISBNs are not available, their titles and authors could be compared. The system also needs to list the prices offered by each site. Thus, the system needs to know the semantic of each data unit. Unfortunately, the semantic labels of data units are often not provided in result pages. For instance, no semantic labels for the values of title, author, publisher, etc., are given. Having semantic labels for data units is not only important for the above record linkage task, but also for storing collected SRRs into a database table.

Proposed System:

In this paper, we observe that, for an important class of queries over news archives that we call time-sensitive queries, topic similarity is not sufficient for ranking. For such queries, the publication time of the documents is important and should be considered in conjunction with the topic similarity to derive the final document ranking.

Most current methods for searching over large archives of timed documents incorporate time in a relatively crude manner: users can submit a keyword query and restrict the results to articles written sort the results on the publication date of the articles. Unfortunately, searchers do not always know the appropriate time intervals for their queries, and placing the burden on the users to explicitly handle time during querying is not desirable. So that to overcome the burden by grouping related documents in a single group.

The grouping documents can be done by using clusterization algorithm. In this paper we are using k-means clustering algorithm for grouping same related documents. Before perform the clusterization of document you can perform searching operation for query related documents. After retrieving document we can find file relevance for each document. Before finding file relevance of document we can also count query matching words in a particular document. After finding of file relevance we can perform the clusterization of documents based on publishing time of documents.

We propose an efficient search implementation with respect to frequency manipulations and time stamp for efficient results for user specified query. We integrated traditional relevance score method and time stamp approach. In this approach data owner out sources the data in the server, before storing data in the server. Data owner has a collection of data samples that he wants to outsource on the server where several important features are selected and extracted. The extracted data is given to any one of the clustering algorithm wherein we use partition aroundmedoids (k-medoid) algorithm. The clustering algorithm clusters the data set according to publication date.

Hence, when a query is submitted the query is processed according to the time intervals formed by the clustering and the result is interpreted. The most common realization of k-medoid clustering is the Partitioning around Medoids (PAM) algorithm. The PAM method first computes representative objects called as medoids. A medoid can be defined as an object ofcluster whose average dissimilarity to the other objects in the cluster is minimal. After finding the set of medoids the other objects are put into and is as follows:

1. Initialize: randomly select k of the n data points as the medoids.

2. Associate each data point to the closest medoid. ("closest" here is defined using any valid distance metric, most commonly Manhattan Distance)

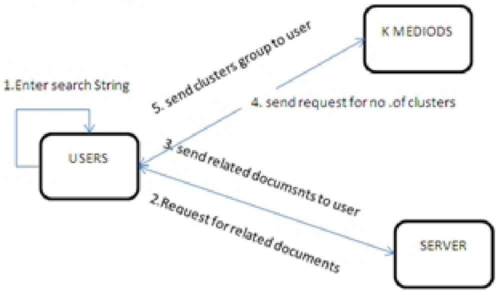
3. For each medoid m
a. For each non-medoid data point o
b. Swap m and o and compute the total cost of the configuration

4. Select the configuration with lowest cost.

5. Repeat steps 2 to 4 until there is no change in the medoid.

$$\mathrm{cost}(x,c) = \sum_{i=1}^{d} |x_i - c_i|$$

Architecture diagram:


K MEDIODS
1.Enter search String
5. send clusters group to user
4. send request for no .of clusters
USERS
3. send related documsnts to user
2.Request for related documents
SERVER

CONCLUSION:

Now a days 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_mediods 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] W. Dakka, L. Gravano, and P.G. Ipeirotis, "Answering General Time-Sensitive Queries," Proc. 17th ACM Conf. Information and Knowledge Management (CIKM '12), pp. 1437-1438, 2012.

[2] M.MiladShokouchi,KiraRadinsky, "Time Sensitive Query Auto completion" Proc .35th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval,2012.

[3] M. Shokouhi. Detecting seasonal queries by time-series analysis. In Proc. SIGIR, pages 1171-1172, Beijing,China, 2011.

[4] Zhumin CHEN1, Haichun YANG1, Jun MA1, ,Jingsheng LEI2, Haihui GAO1," Timebased Query Classification and its Application for Page Rank, Journal of Computational Information Systems 7: 9 (2011) 3149-3156

[5] S. Chaudhuri and R. Kaushik.Extending autocompletion to tolerate errors. In Proc.SIGMOD,pages 707-718, Providence, Rhode Island, USA, 2009.

[6]S. Bickel, P. Haider, and T. Sche_er. Learning tocomplete sentences. In Proc. ECML,pages 497-504.2005.

[7]J. Fan, H. Wu, G. Li, and L. Zhou. Suggestingtopic-based query terms as you type. InProc.APWEB, pages 61{67, Washington, DC, 2010.

[8]Mani, I. and Wilson, G. 2000. Robust temporal processing of news. In ACL '00:Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, USA, 69-76.

[9]J. L. Elsas and S. T. Dumais.Leveraging temporal dynamics of document content in relevance ranking.In Proc. WSDM, 2010.

[10] R. Cleveland, W. Cleveland, J. Mcrae, and I. Terpenning. STL: A seasonal-trend decomposition procedure based on loess.Journal of O_cial Statistics, 6(1):3-73, 1990

[11]ZeynepPehlivan, Anne Doucet, StephaneGancarski :Changing vision for access to web Archives, TWAW 201, March 28,2011, Hyderabad,India.

Author Details :

kadambala sankar is a Student in M.Tech(SE) in Sarada Institute of Sci¬ence Technology And Management, Srikakulam. He Re¬ceived his B.Tech(CSE) prajna institute of technology and management, palasa,JNTU Kakinada Andhra Pradesh.His interesting areas are Network,data mining.

Chintada Sunil Kumar working as a Asst Professor of CSE in Sarada Institute of Science, Technology and Management (SISTAM), Srikakulam, Andhra Pradesh. He received his M.Tech (CSE) from Jntuk, Kakinada. Andhra Pradesh. His interest research areas are Database management sysytems,Computer Architecture, Image Processing, Computer Networks, Distributed Systems. He published 4 international journals and he was attended number of conferences and workshops