Research Journal of Applied Sciences, Engineering and Technology 10(3): 267-273, 2015
DOI: 10.19026/rjaset.10.2487
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: December 14, 2014 Accepted: February 8, 2015 Published: May 30, 2015

Research Article

Equipped Search Results Using Machine Learning from Web Databases

[1]Ahmed Mudassar Ali and [2]M. Ramakrishnan
[1]Bharath University, Chennai, India
[2]School of Information Technology, Madurai Kamaraj University, Madurai, India

Abstract: Aim of this study is to form a cluster of search results based on similarity and to assign meaningful label to it Database driven web pages play a vital role in multiple domains like online shopping, e-education systems, cloud computing and other. Such databases are accessible through HTML forms and user interfaces. They return the result pages come from the underlying databases as per the nature of the user query. Such types of databases are termed as Web Databases (WDB). Web databases have been frequently employed to search the products online for retail industry. They can be private to a retailer/concern or publicly used by a number of retailers. Whenever the user queries these databases using keywords, most of the times the user will be deviated by the search results returned. The reason is no relevance exists between the keyword and SRs (Search Results). A typical web page returned from a WDB has multiple Search Result Records (SRRs). An easier way is to group the similar SRRs into one cluster in such a way the user can be more focused on his demand. The key concept of this paper is XML technologies. In this study, we propose a novel system called CSR (Clustering Search Results) which extracts the data from the XML database and clusters them based on the similarity and finally assigns meaningful label for it. So, the output of the keyword entered will be the clusters containing related data items.

Keywords: Annotation, clustering, data wrappers, web database, XML, XML data extraction, XQuery

INTRODUCTION

More databases that are accessed through web are termed as Web Databases (WDB). Such databases can be accessible to the outside world of end users in abstracted manner via HTML form based interfaces. These interfaces are intended in providing the connectivity between the system and the user. Every result page that is generated by the system contains more number of search results in record format. Each result of the page is highly unique in nature that provides dedicated information. Clustering of search results organize the results returned by topic, thus providing an alternative view of hierarchy which is totally different from the traditional search.

The hierarchy of cluster is highly advantageous because:

- The similar results can be clubbed together and they are kept under one cluster. So dissimilar results will be separated.
- It avails us with better topic understanding by providing high level overview of the results returned.
- It allows accurate summarization of the search results.
- It can be considered as an added feature to the conventional search.

It is necessary for the users to know only the content which are actually needed by them and rich in information. The users need not know the logics behind and the deployment data, where from the information is retrieved, the partitions of the retrieved data etc. This scenario can be well implemented using three tier architecture of web deployment. Through web databases concept, the data stored in one system can be accessed from any remote system with internet facility. This indeed reduces the burden of having multiple data copies in many places where the access is synchronized inbuilt. This will be more useful for businesses and domestic purposes. Another advantage in using Web Databases is there is no special software is needed to use it in this modern wi-fi enabled power packed hand held devices era. The necessary thing is a system or tablet with a standard browser. As the demand for data has been increased, the numbers of Web Databases also have been increased.

Information Extraction (IE) is an automated process of extracting information from documents. The user query is in natural language. It is converted into technical query by means of a defined process called Natural Language Processing (NLP). It is accomplished by machine learning techniques and using the query utility.

Machine Learning concepts are the part of Artificial Intelligence (AI) concepts that learns the

Corresponding Author: Ahmed Mudassar Ali, Bharath University, Chennai, India

behavior of the system from the past working and preserve it for future.

This study has following contributions:

- While the conventional searches simply extract the information and display it, we thoroughly analyze it and cluster the similar results together and the dissimilar separated.
- The clustering activity is taken care not solely by the logics. It is shared with the XQuery utility too. Thus the burden on clustering is divided into two viz., XQuery for partial clustering and bean component for cluster organization. Finally, the clustering turns into full fledged functionality.
- The database for retrieving is made as extensible Markup Language (XML). It can consume large volume of data with smaller storage area; hence it is hierarchical in nature and semi structured.
- The most suitable label is assigned to every cluster created. It makes the search results meaningful in nature for better understanding.

Information Extraction and Semantic Indexing have been active research areas for decades. Achieving semantic indexing has different perspectives. One is through Ontology concepts where as other is grouping Annotating Search Results from Web Database (Lu *et al*., 2013) proposed an approach for annotating the retrieved results in better understanding format. A dedicated alignment algorithm has been used to align the content and for handling text nodes and data units in a page. Generally a search result page contains multiple search result records in a format that combines text nodes and data units altogether. A wrapper induction system is used to hold all the extracted data out in it temporarily before aligning them accordingly. Different annotators are used at multiple levels for annotating it perfectly.

Features extraction algorithm from sgml for classification (Abdullah and Hitam, 2007) proposed the basic phases in text categorization include preprocessing features, extracting relevant features against the features in a database, and finally categorizing a set of documents into predefined categories. An algorithm for pre-processing features seems to be like a "black-box" and ignored by them. Thus, it is significant and worthwhile to develop an algorithm for preprocessing features and finally can be used by other beginners before going in depth in the field of text categorization.

Clustering method: Search result Based web personalization (Avinash *et al*., 2012) proposed to mine a reduced set of effective search result for enhancing the searching experience. They store and maintain user's long-term dynamic profile based on user search and use it to personalize. Ontology is used at client side to solve the cold start problem and expand the query and generate clusters of similar results. Client's profile is stored as a weighted ontology tree. Web search results from an existing search engine are taken and re-rank them based on client's profile.

SR cluster: Web Clustering Engine based on Wikipedia (Meiyappan *et al*., 2012) proposed a new web clustering engine named SR Cluster to overcome deficiencies, in specific for the polysemy unigram search keywords. SR Cluster identifies the possible categories and its label for the given polysemy keyword based on Wikipedia.

Learning to Cluster Web Search Results (Zeng *et al*., 2004) in this research they reformalize the clustering problem as a salient phrase ranking problem. Given a query and the ranked list of documents (typically a list of titles and snippets) returned by a certain Web search engine, their method first extracts and ranks salient phrases as candidate cluster names, based on a regression model learned from human labeled training data.

A Model for XML Schema Integration (Passi *et al*., 2002) proposed an object-oriented data model called XSDM (XML Schema Data Model) and present a graphical representation of XML Schema integration. The three layers included are, namely, pre-integration, comparison and integration. During pre-integration, the schema present in XML Schema notation is read and is converted into the XSDM notation. During the comparison phase of integration, correspondences as well as conflicts between elements are identified.

Beyond Single-Page Web Search Results (Varadarajan *et al*., 2008) proposed a technique that given a keyword query, on-the-fly generates new pages, called composed pages, which contain all query keywords. The composed pages are generated by extracting and stitching together relevant pieces from hyperlinked Web pages, and retaining links to the original Web pages.

Web Usage Mining Based on Complex Structure of XML for Web IDS (Eshaghi and Gawali, 2013) proposed a technique that will present how various web log files in different format will combined together in one XML format to further mine and detect web attacks. And because log files usually contain noisy and ambiguous data this study will show how data will be preprocessed before applying mining process in order to detect attacks.

XML with Cluster Feature Extraction For Efficient Search (Divya *et al*., 2013) they proposed that the Data Owner can upload the Documents from any Database Format So that it is converted into XML Format.

Optimization of Web Content Mining with an Improved Clustering Algorithm (Bisht and Bansal, 2013) proposed a method to improve the use of advance web data clustering techniques which is highly used in the advent of mining large content based data set which allows data analysts to conduct more efficient execution of large scale web data searches.

Improving Finding and Re-finding Web search Results Using Clustering and Visualization (Badesh and Blustein, 2011) proposed a Data Mountain Search

Results Presentation Interface (DMSRPI) and a plan for a user study of its effectiveness. The interface is intended to improve the effectiveness and efficiency of users' search of the Web.

Annotating Structured Data of the Deep proposed an approach of encoding every data into the result page and they are returned for human browsing. Multi annotator approach classifies the extracted data at different levels to organize the content in a grouped manner that doesn't deviate the original sense. All in one news (www.allinonenews.com) is the domain working under this concept. The concept fairly works but with limited amount of data. When data becomes enormous in nature, the relational data extracted out doesn't accommodate in the space provisioned. Record Expression (REXP) is the framework used to withhold the data.

Richi Nayak, Rebecca Witt and Anton Tonev proposed 'Data Mining and XML Documents' (Nayak *et al*., 2002) proposed the data mining via XML documents can be achieved through clustering DTDs and corresponding schemas. While clustering such XML documents tags provide the semantics of the data being surrounded in it. It suggests more clustering techniques such as content clustering, structural clustering. Among those concepts, structural clustering is used as it matches with our concept.

'Effective Web Data Extraction with Standard XML Technologies' (Myllymaki, 2001) proposed an architecture call ANDES framework. The idea is expanding the simple screen scraping. The paper provides a wide knowledge in dealing with XML and XSL technologies for web service concepts. Batch oriented data extraction called crawling is achieved for deep web extraction. Our focus is towards the content mining via XQuery technology.

Toolkit for generating wrappers (Kuhlins and Tredwell, 2003) has provided an analysis of the available toolkits in the market. It avails a fair overview of the merits and demerits of the all possible tools thus we decided to use a new tool which has not been used so far while dealing with web search area. The tool is exist database server.

However, the key concept of the paper is XML technologies. Using xml as backend, the details that match with the user query are taken out based on exist database system using the available interfaces and using the dom parser. Using exist the dedicated xml is stored and used for querying purpose (Fig. 1).

Existing technique presents a solution to provide the search results from the web search engines that contain a separate dedicated database or from many databases that are publicly available. It provides the results in an annotated form so that the user can better understand about the query being searched. i.e., along with the necessary content, additional information will be displayed to user by making the result page rich in information (Fig. 2).

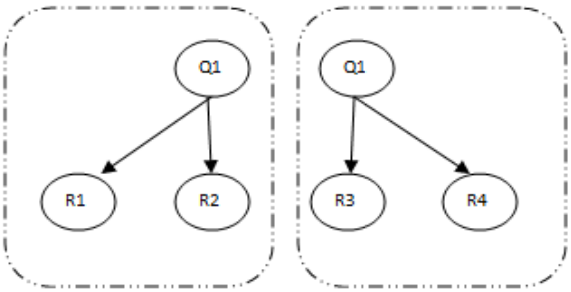

Fig. 1: Example to show how results are related to queries

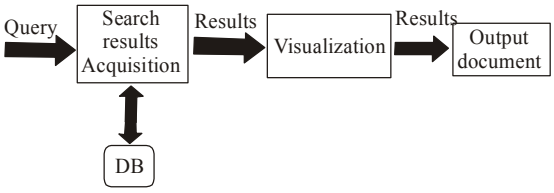

Fig. 2: Overall architecture of data extraction process

Some of the issues being occurred in the conventional search are:

- The search results act individually in the result page which leads the user getting deviated from the search content.
- More repetition exists in the result page.
- Annotation is done at search result level.

In this process, the query is received from the user. It is taken for processing. The system searches the query in the underlying database. The results are converted into a presentable format and they are shown as the result page for the corresponding query.

MATERIALS AND METHODS

An easier way is to group the similar SRRs into one cluster in such a way the user can be more focused on his demand. Thus some more deviations can be omitted. In this study, we propose a system called CSR (Clustering Search Results) which extracts the data from the XML database and clusters them based on the similarity and finally assigns meaningful label for it. So, the output of the keyword entered will be the clusters containing related data items.

The system takes the input from the end user via the interface. The system takes backend input as the XML document. The user provides the search query through a Web User Interface of any specific application. The query will be searched in the underlying XML database for its presence. If it is present, our proposed system comes into picture. The XML database will be considered and the keyword based extraction is done. Finally a featured XML containing the user's query related information is generated.

The resulted XML is taken into consideration. The core part of the proposed system called 'clustering' is being performed on the XML data. The clustering is done by XQuery utility which is a dedicated querying capability of XML and the rest of the cluster organization is done by java's API.

Finally the grouped data is assigned with a meaningful label. Annotations are done at both cluster level and item level. The search results are the pure clusters. The raw search results can be viewed by getting into the cluster accordingly.

The system computation is realized through the application of three main software modules implemented in Java. Each module exploits results coming from the execution of the previously applied module. This system does not need the human intervention during automatic extraction process. This extraction process adopt exist database technology for information extraction. A featured XML based on the user input is created with XQuery capability.

The overall process has been divided into three core module as follows:

- XML Data Extraction
- Clustering
- Annotation

All the three modules are logically split. But indeed, they have been implemented in parallel to handle huge amount of data. Thus the complexity in storing the data temporarily is completely avoided.

XML data extraction: It involves the following sub modules viz.:

- Tokenization
- Data extraction

As the data extraction module deals with the database, exist database server is used for maintaining the XML database that serves as backend.

Tokenization: The input is received as natural language using JSP/SERVLET concepts. This is established using Apache Tomcat web server. The Java Server Page holds the interface for user communication. Once the user enters the keyword as input, the corresponding servlet container starts its execution and receives the input as a whole. The keyword entered by the user is then tokenized using Java language with its dedicated API. This tokenization module facilitates the extensibility to search all the relevant search results available at looks and corners of the XML document.

Data extraction: Here, the preliminary work of preparing the data for processing is not needed because of the semi structured format of the nature of XML data. The XML document that serves as backend is considered as a whole once the input is received. It is situated in exist database server location that is one of the high performance native XML engines. It is fast in nature as it deviates with its working nature from traditional SQL queries. Alternatively, it uses XQuery technology to query the underlying XML.

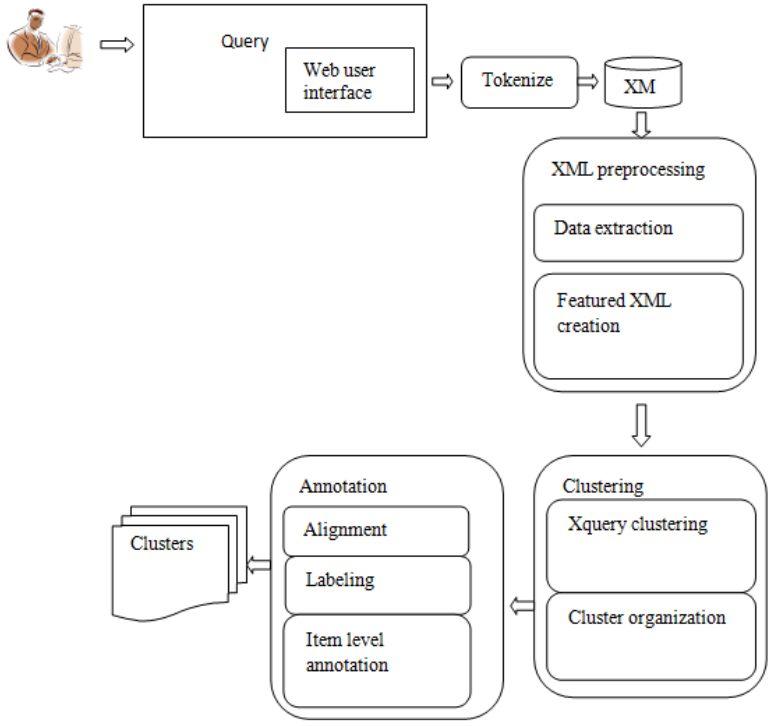

Fig. 3: System architecture of clustering search results

Thus the XQuery extracts the relevant data present in the XML that match with the tokenized keywords. This module points to the location of the XML document. The pre setups like the designated port for the database server (here, it is 8080), the collection path, resource and the drivers are initialized. An instance is created for the database. Then the collection is accessed via the database manager where the XML database is registered. Later the XQuery service is accessed through the collection. The results returned by the XQuery are stored in a Resource Set type of instance which is connected to the XQuery service (Fig. 3).

Clustering: This module involves two sub-modules viz.:

- XQuery Clustering
- Cluster Organization

XQuery clustering: Once the XQuery fetches the data, they are grouped together based on the similarity (i.e.,) by matching of the keyword with XML content thereby grouping the similar contents together which is achieved with the capability of XQuery. The entire relevant records are extracted from the original XML document with its structure possessed.

Cluster organization: The data along with the grouped nature are received by Java again. Java has the 'Result Set' type of collection variable that is capable to store a large set of data in its vector variable. It identifies the distinct data in the groups of data sent by XQuery. Thus it organizes the similar data into one cluster and performs a number of servlet executions to organize the corresponding data that belong to a single cluster.

Annotation: Literally, the term 'annotation' means providing the information in more presentable format for better understanding and with an improved user interface. Here, annotation is established in multiple levels.
This module involves:

- Alignment
- Cluster level Annotation/Labelling
- Data item level Annotation

Alignment: The cluster containing the data element is sent for further processing to the alignment module regarding their placement. This captures the attributes of every SKUs present in the clustered extracts of XML data. It performs a study on them and it prepares where to place the data, which attribute has to come in which position in the results page etc.

Thus it generates a template containing the spaces to store the data elements. This template is filled by all the SKUs with their own values. Thus uniformity is maintained in the data display. Available frameworks and templates are used to align the data for accommodation.

Cluster level annotation/labelling: Once the data is aligned, every cluster has to be identified with a descriptive keyword indicating what it contains. This is accomplished using labelling. As the label should be more relevant and highly suitable to the data items, the label here chosen is the category under which the data items are present. Thus a html page is created and it is referenced as the cluster naming with the label assigned. So, finally the output will be the clusters that are serving as http references.

Data item level annotation: As the final step, all the data are aligned and annotated with its attributes. The similar data are kept in a single html page and they are displayed in descriptive manner. Thus the extracts are spitted and combined based on its similar nature and annotated. A syntactic clustering event is handled in XQuery and the data is then organized using java APIs. Later they are annotated at two levels viz., cluster level and data item level. Finally fully fledged html pages are created, each containing similar data items. The pages act as the containers of the clustered extracts of the search results.

Algorithm: The entire strategy is summarized in below Algorithm 1 and a sub strategy is depicted in Algorithm 2.

Algorithm1: Clustering Framework
Input: A set of search results SRs returned from web
Output: Clustering of SRs into clusters C.

1: Initializations: C = 0; uI [x] = 0 where x is an iterative variable; x = N where N is the initial assumptive of clusters C; index = 0.
2: uI = user Input;
3: ds = getResultSet (uI)
4: HashMapobj = result [N];
5: Repeat
6: For each x€uI and ds is null
7: Do
8: Query uI against the item x;
9: Update x in HashMap<obj>
10: End for
11: x++;
12: Until x< = N
13: Return ds

Table 1: Characteristics of datasets

Data sets	# of classes	# of features	# of examples
Laptops	6	4	20
Tablets	2	3	36
Pen drives	3	2	12
Smart phones	5	5	44
Speakers	3	2	32
Monitors	4	3	11

Algorithm 2: getResultSet (uI)
Input: search query in natural language
Output: Web search results:
1: Initializations: uI = null; builderFactory = instance; result [N] = null;
2: Repeat
3: X* = tokens of uI;
4: For each i€uI [N];
5: Query x* at every level;
6: Update the Result [i]
7: i++;
8: End for;
9: Until i<element.length;
10: Return result [N];

The above stated algorithms perform the data extraction and the clustering in parallel. The clustering activity is done in recursive manner. The module is self recursive and the recursion persists until it finds no other items left for clustering from the database.

RESULTS AND DISCUSSION

The input is the keyword in natural language entered by the user. The results extracted and returned purely depend upon the backend database contents and the user input. So the input will be the keywords indicating the products available in the retail store. Database Server Collection Path is the server location of exist database. The server is available in the path with admin privilege as admin@xmldb: exist:// localhost: 8080/ exist/xmlrpc.

The database is available in the sample folder of above path. It provides the command level utility on the database access. Thus any search at admin level can be made easily for the changes to be made on it. The permissions can be set accordingly by admin level security. The XQuery capability is achieved via servlet. The querying connections are made before servicing the query. The retrieved results are clustered using the corresponding business logics. The clustered data items are labeled at cluster level in syntactic manner (Table 1).

Above is the analysis observed on the whole for different datasets mentioned. The active learning was done by the clustering framework developed.

CONCLUSION

Building a system for clustering the search results returned from the web database is challenging task. Web databases have been frequently employed to search the products online for retail industry. Most of the times the web databases used for this purpose are the dedicated ones and private to the retailer/concern. Whenever the user queries these databases using keywords, most of the times the user will be deviated by the search results returned. The reason is no relevance exists between the keyword and SRs (Search Results). A typical web page returned from a WDB has multiple Search Result Records (SRRs). An easier way is to group the similar SRRs into one cluster in such a way the user can be more focused on his demand. In this study, the proposed system CSR (Clustering Search Results) extracts the data from the XML database and clusters them based on the similarity and finally assigns meaningful label for it. So, the output of the keyword entered is the clusters containing related data items. Thus the system achieves the clustering from the syntactic point of view. Existing work done in the field of clustering is achieved with the XQuery capability cum java APIs.

REFERENCES

Abdullah, Z. and M.S. Hitam, 2007. Features extraction algorithm from Sgml for classification. J. Theor. Appl. Inform. Technol., 3(2): 72-78.

Avinash, A.P., P.M. Narayankar and M.M. Jeevitha, 2012. Clustering method: Search result based web personalization. Proceeding of International Conference on Advances in Computer and Electrical Engineering (ICACEE'2012), pp: 94-97.

Badesh, H. and J. Blustein, 2011. Improving finding and re-finding web search results using clustering and visualization. Int. J. Intell. Comput. Res. (IJICR), 2(1/2/3/4): 228-234.

Bisht, S.S. and S. Bansal, 2013. Optimization of web content mining with an improved clustering algorithm. Int. J. Emerg. Technol. Adv. Eng., 3(11).

Divya, D., A. Muthukumaravel and P. Mayilvahanan, 2013. XML with cluster feature extraction for efficient search. Int. J. Emerg. Technol. Adv. Eng., 3(8): 292-295.

Eshaghi, M. and S.Z. Gawali, 2013. Web usage mining based on complex structure of XML for web IDS. Int. J. Innov. Technol. Explor. Eng., 2(5): 323-326.

Kuhlins, S. and R. Tredwell, 2003. Toolkits for generating wrappers-a survey of software toolkits for automated data extraction from web sites. In: Aksit, M., M. Mezini and R. Unland (Eds.), NODe 2002. LNCS 2591, Springer-Verlag, Berlin, Heidelberg, pp: 184-198.

Lu, Y., H. He, H. Zhao, W. Meng and C. Yu, 2013. Annotating search results from web database. IEEE T. Knowl. Data En., 25(3): 514-527.

Meiyappan, Y., N. Ch. S. Narayana Iyengar and A. Kannan, 2012. SRCluster: Web clustering engine based on Wikipedia. Int. J. Adv. Sci. Technol., 39: 1-18.

Myllymaki, J., 2001. Effective web data extraction with standard XML technologies. Proceeding of 10th International Conference on World Wide Web (WWW, 2001). Hong Kong, pp: 689-696.

Nayak, R., R. Witt and A. Tonev, 2002. Data mining and XML documents. Proceeding of International Conference on Internet Computing (IC'2002). Las Vegas, Nevada, pp: 660-666.

Passi, K., L. Lane, S. Madria, B.C. Sakamuri, M. Mohania and S. Bhowmick, 2002. A model for XML schema integration. In: Bauknecht, K., A.M. Tjoa and G. Quirchmayr (Eds.), EC-Web 2002. LNCS 2455, Springer-Verlag, Berlin, Heidelberg, pp: 193-202.

Varadarajan, R., V. Hristidis and T. Li, 2008. Beyond single-page web search results. IEEE T. Knowl. Data En., 20(3): 411-424.

Zeng, H.J., Q.C. He, Z. Chen, W.Y. Ma and J. Ma, 2004. Learning to cluster web search results. Proceeding of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04), pp: 210-217.