*International Journal of Computer Applications (0975 – 8887)*
*Volume 52 – No. 8, August 2012*

# Web Supported Query Taxonomy Classifier

Asiya Abdus Salam Qureshi
Department of Computer Science and Software Engineering
University of Hail, Saudi Arabia

Syed Muhammad Khalid Jamal
Department of Computer Science
University of Karachi, Pakistan

## ABSTRACT

A new approach called Web Supported Query Taxonomy Classifier is introduced in this paper, which generates better searching results. We combine WSQ, Web Supported Queries approach with Query Classification in which bridging classifier and category selection method is used for classification of queries. In this paper, Query categorization will build bridging classifier in an offline mode at mediator taxonomy integrated with the category selection method for the effectiveness and efficiency of online classification and then classifier is used for mapping input queries to target taxonomy in online mode. After this the queries are settled in virtual tables presented by WSQ tuples for the generation of web search result calling one or more search engines for improved results. Further a technique called asynchronous iteration is used for concurrency among multiple web search requests and query processing.

## Keywords

Web Query Classification, Asynchronous Iteration, Bridging Classifier, Category Selection Method, WSQ, Web Supported Query Taxonomy Classifier.

## 1. INTRODUCTION

Nowadays information is available on single click. Internet acts as a tool which contains immense information. Web users input their required data and search engines search those queries. A number of processing phases are included in a very basic search engine, it includes, Indexing, Crawling, Query Processing, and Ranking. Indexing: to build an index to facilitate query processing, crawling: to discover the web pages on the internet, Query Processing: To Extract (based on user's query terms) the most relevant page and Ranking: Based on relevancy order the result.

Real time information is maintained by the search engines based on algorithm running on a web crawler. A web crawler is an agenda for computer searching. It's a computer program that browse the world wide web in an automated or systematic way. Many search engines use web crawling as a way of providing up- to- date data. A copy of all the visited pages is created for later processing by a search engine that will make a directory for the downloaded pages to provide fast searches.

Search engines have gained popularity over the World Wide Web. Information moves between the World Wide Web which is considered un-structured data and database which is structured data. Figure 1 shows the relationship of structured and unstructured data which provides vast amount of information. Here split of information between huge amount of unstructured information available over World Wide Web and structured data stored in databases is depicted. Traditional object oriented, relational, and object relational databases operate over well controlled typed data and language such as OQL and SQL queries.

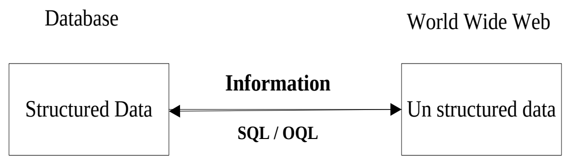


***Figure 1 Information Split***

Significant deficiencies with respect to flexibility, precision and robustness are found over web search techniques. They take queries as input because these user queries are important medium by which a system can understand user's interest. These queries are assorted and significant which requires classifying them into small commercial taxonomy. They are short so they are vague; therefore they lack precise semantic description.

Query classification is an application for giving better search pages result and WSQ arrange and manage data between structured and unstructured information available on internet. After the query input, query categorization comes to action which results in multi-classification, because same query may be classified under different key words. Due to the reason that, it can be contained as in input in more than one target category, it is imprinted to an intermediate category and then applied to the target category. With the help of bridging classifier, mediator taxonomy in offline mode is connected to target categories in online mode. This reduces the creation of classifier every time if changes are made in target categories.

A category selection method is applied for selection of categories in mediator taxonomy. This help to improve effectiveness and efficiency of online classification. Further, we use practical way for managing the searches. WSQ results in producing relevant web search results in structured way.

WSQ (Web Supported Queries) referred to an approach over a relational database which control result from web searches to enhance SQL queries. Two virtual tables enabled the queries supported by WSQ whose tuples represents web exploration outcomes generated during query execution one or more search engines that involves many high latency calls, forces the workstation to become inoperative.

A technique known as asynchronous iteration is used for linking multiple search requests and query processing in parallel. Web query classified request will be passed on to WSQ in which category selection mechanism and asynchronous iteration will work for improving relevant search.

Query taxonomy is the organization of search results which are clustered into hierarchical structure. This is an easy, convenient and comprehensive way for user to browse relevant web contents. When the categorized query results are supported by WSQ, two virtual tables WebPages and Web Count are created. WebPages are unbounded table that contains all the URLs that search engine returned. Web Count is a comprehensive view of WebPages for counting total number of URLs returned.

In order to achieve competence, a query processor can issue many web requests concurrently. Asynchronous iteration attains low transparency for external virtual table access and it is incorporated into conventional relational database system. Asynchronous iteration can be defined as a method that enables a single query, executed by non parallel relational query processors in such a fashion that multiple concurrent web searches can be done.

Figure 2 shows the basic query classification and WSQ architecture. Each WSQ instance queries are routed in one or more databases after passing through classified bridge. Each query is classified with the help of query categorization and then moves to database after the synchronized execution of queries.

Organization of this paper is as follows. Section 1 contains introduction with detail overview. In section 2, some related work is discussed along with the methods used in the paper. In section 3, the probabilistic framework and the processing of Web supported query taxonomy classifier is discussed. Section 4 is about proposed algorithm which is the main target of this paper. Section 5 presents methodology with example. Lastly section 6 concludes the work with future work.

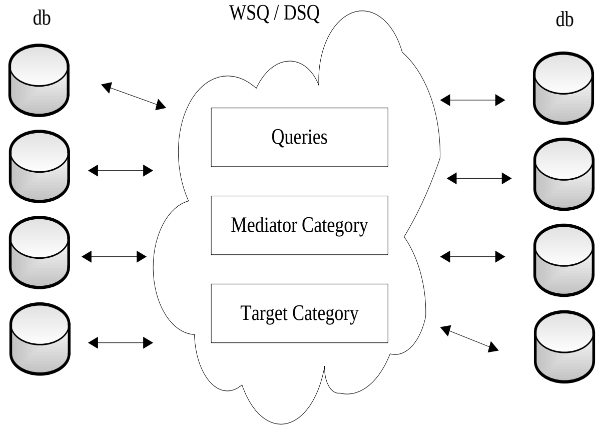


***Figure 2 Basic Architecture of Web Search Query Taxonomy Classifier***

## 2. RELATED WORK

The work related to this new approach is based on building bridges for web query classification and Web Supported Query. Web user queries are classified into target taxonomy by Shen et al. Some of similar wok has been done by Gravano et al [20] and Beitzel et al [21]. All their work is on classification of queries but the major difference between our approach and others is that the trained data is used for categorization of the result required. This is the extended form. KDDCUP 2005 [6] results are under taken for summarized clarification. Also a practical approach of WSQ/DSQ for combined querying of databases and the web is used that help in supporting the search engines on web. Wrappers give useful information for web pages as database contents. Semi structured data models like XML structured the wed data and queries. Our work differs in a way that we utilize the techniques that web search result in structured and taxonomical form. Furthermore, the papers of CDY95 [11] and DM97 [18] are also related for WSQ/DSQ but with few differences like minimizing the numbers of external calls, assumption of external text sources return search results as unordered or optimizations are not useful under the use of languages. The previous work on dependent joins can be seen in query optimization algorithm provided in FLMS99 [19]. Our search is on merging the issues left in WSQ/DSQ and query classification and creating relationship between targeted search queries and taxonomies.

## 3. COMPARATIVE ANALYSIS

Internet acts as an important means containing increasing information for web searchers who submit their request in the form of short web queries for searching. These unclear queries create problem of query classification in terms of set of target categories. Following sections shows the related work of this paper.

## 3.1 Query Classification

QC algorithm grouped the users' categories. No result will be produced if there is no matching between target categories and queries. Sometimes, due to evolution in web contents, target categories have to change. From KDDCUP 2005 [6], it was observed that 67 target categories were used to classify 800,000 real web queries. There is innovative method for mapping queries to target categories. Mapping of input query to intermediate category and then to target category takes place.

The Query Classification algorithm uses bridging classifier on an intermediate taxonomy in an offline mode and then uses the online mode to plot queries to target categories. For improving the online classification, category selection method is used to select intermediate taxonomy categories. There is no need for extra resource as well and the method proves to be more accurate. Inflexible definition of QC problem includes classifying a user query into a ranked list of n categories. Each output is ranked higher than the other. When web users submit queries, its meaning and intension are subjective. The target categories look like hierarchical tree, each node of which represents a single category. Semantics of each node is defined by labels which lead from root to the corresponding node.

One straightforward method to get related pages is to submit each of them to web search engines and get the queries from the target categories which can provide us with semantics of the categories and queries. Features should be considered which must include, full plain text of a page, the snippet generated by search engines and title of a page with the html tags removed. The word matching between categories can be done using method called extended matching in which expand the words in the labels of target taxonomy through vocabulary. Our data will be searched in target taxonomy. An equation is used to show the mapping of intermediate taxonomy on to target taxonomy. In the equation, indicator function is used having values o and 1 accordingly and category in the intermediate taxonomy for marking the page returned after search. There will be low recall if the intermediate taxonomy does not match the search within the target taxonomy.

Query classification method helps in the betterment of low recalls with Support vector machine. Three steps improve the performance, first, after the training data creation, intermediate category is mapped to target category. Second, SVM is trained for target categories. Third, search engine is used to get enriched result with the help of SVM classifiers. Intermediate taxonomy is taken as bridge for linking target taxonomy and queries. We use here another equation in which conditional probability of target category is used which helps to find the possibility of the presence of the search in the target category.

Due to enormous categories and irrelevancy of queries, reduction of computation complexity is done by category selection. For this purpose, two approaches, Total Probability and Mutual Information are used. Total probability gives score according to its probability to each intermediate category. Mutual information is used in statistical language modeling of word association. In the experiments carried out, examples for data sets are taken from KDDCUP 2005 [6]. These datasets act as validation data. 800,000 queries are taken from MSN search logs for the solution. Three human query labelers L1, L2 and L3 are used with Precision and F1 score values. The average values among these three are formed to be 0.5 which mean problem of query classification is a daunting task.

The evaluation measurements are based on precision, performance and creativity for the evaluation of submitted solution. Creativity is based on the solution judged by experts and other two evaluate performance, precision, recall and F1. On the bases of Results and Analysis, the performance depends on exact matching and SVM. Number of documents gathered for target categories and after expanding the category presents each element. Then performance of exact SVM by F1 and matching and precision is done. Analysis is done by mapping direct matching and extended matching with SVM and exact matching. The result appeared to be better performance of Exact E and SVM E than Exact D and SVM D respectively. Also Exact E performs better than SVM E. The use of thesaurus is important that help expand the key-words in tags of target categories for the improvement of mapping function but due to some limitations on it, bridging classifiers are used. The best performance by bridging classifiers is when n equals 60. The cost of computation will increase if number of intermediate categories increase but the result achieved shows Mutual Information performance is better than Total Probability.

## 3.2 WSQ/DSQ

OQL and SQL enable expressive ad hoc queries, while keyword based, fairly simple queries are supported by search engines. Search engines are being exploited by WSQ/DSQ to augment SQL queries over a relational database and WSQ/DSQ uses a database to explain and enhance searches over the web. Users are enabled by WSQ to write intuitive queries over SQL that automatically executes web searches relevant to the query and search results are combined with the structured-data in the database manual searching. Web keyword searches are enhanced by DSQ with relevant information content in the existing database. DSQ uses the web to correlate that phases with the terms in the recognizeddatabase.WSQ is based on introducing two virtual tables, web pages and web count to any relational database. A single query that is being executed by parallel relational query processors is enabled by Asynchronous Iteration method to hold multiple concurrent Web searches. Wrappers make information content available in web pages act rationally as database elements by enabling expensive queries over data and facilitating the assimilation of data/information content from various sources.

A query redraft scheme was presented for automatically converting queries that call a search engine via a user defined interface and converts queries into more proficient queries that integrate a virtual table content based on search engine exploration. Dependent join, supply bindings to virtual tables when web searches are integrated into a SQL query. Search Expression is used which is a parameterized string representing a web search expansion. Web pages containing 0 or more records where Rank, attributes URL, and date are the values returned back by the search engine has a rank which equates to 1. A virtual table Informix scans the associated access where conditions can process the selection. Optimizations are proposed by CDY95 [11], reducing the number of caching techniques and external calls. A logical option to evaluate is a Volcano. For performing all 5D searches concurrently, not only the parallel query processor must dynamically partition the problem in the right way, the 5D query there also must be launched.

Applicability of asynchronous iteration applies where queries depend on values provided by high latency external sources. It is appropriate if many concurrent requests can be handled by an external source or independent calls are issued to many external sources by a query. The query plan generated includes the conversion of EV Scan to AEV Scan and Addition of ReqSync operators to the plan. ReqSync involves Percolation, Insertion, and Consolidation. ReqSync Insertion transformation occurs between its call and blocking operator that waits for its completion. ReqSync Percolation involves the operators that moves up the query plan. ReqSync Consolidation includes two or more than two ReqSync operators that are adjacent now, in the plan. Merging all the adjacent ReqSync operators as they execute the same overall function and multiple placeholders values in a tuples can be managed by a single ReqSync operator.

The implementation and experimental phase notify that red-base is a home grown relational database management system that has integrated the two WSQ virtual tables and asynchronous iteration. It supports a subset of SQL for select project join queries including an iterator based query execution and a page level.

Considerable performance fluctuation may be experienced by the search engine depending on the load and network delays. Due to caching, repeated searches with identical keywords will be accomplished much quickly. Template queries and instantiate multiple versions of them that are structurally alike but result in marginally different searches being performed in order to run many experiments without much delay among each one. Asynchronous iteration can improve the performance of WSQ by 10 or more than 10 as demonstrated by their results. Although, network costs dominate as all the example queries are over very small local tables. The dramatic speed up may not be true for queries involving more complex local query processing.

# 4. OUR WORK: WEB SUPPORTED QUERY TAXONOMY CLASSIFIER

We now describe our new algorithm Web Supported Query Taxonomy Classifier in which queries are connected with target taxonomy by an mediator taxonomy as bridge. Whenever query is input, the required category will match the target taxonomy with the mediator taxonomy. The web query categorization is modeled as way of finding a category ( $C_i$ ) that has maximum probability given a web query (q).

$$Query\ Categorization\ =\ arg\ max\ P\ (\ Ci\ /\ q)$$

If we want to see that a query belongs to given taxonomy or not, uses a Bayesian Transformation

$$P(CiT/q)\ =\ P(q/CiI)\ P(CiI)\ /\ P(q)$$

## 4.1 Probabilistic Framework

If we define the relationship between query q, mediator taxonomy $C_j^I$ and target taxonomy $C_i^T$ using probabilistic framework, then:

$$P(CiT/q)\ =\ \sum P(CiT, CjI\ /\ q)$$

$$=\ \sum P(CiT/\ CjI\ , q)\ P(CjI/q)$$

$$=\ \sum P(CiT/\ CjI\ )\ P(CjI/q) \qquad \text{-- (Eq 1)}$$

Conditional Probability is used to estimate the WebPages in mediator taxonomy, $C^{I.}$ . It is applied to $C_i^T$ when q is given. When we further apply Conditional Probability on above equation, we get

$$P(CiT/q) = \sum P(Ci/\ CjI\ )\ P(q/CjI)\ P(CjI\ )/P(q)$$

Which is equivalent to

$$P(CiT/q)\ =\ \sum P(CiT/\ CjI\ )\ P(q/CjI)\ P(CjI\ )/P(q)$$

Equation 1 can also be written as

$$P(CiT/q)\ =\ \sum P(CjI/CiT)\ P(CiT)\ /P(CjI\ )P(CjI/q)$$

Instead of apply conditional probability to ($C_j^I$/q), apply on $P(C_i^T/\ C_j^I\ )$.

Therefore $P(C_i^T)\ \sum\ P(C_j^I\ /C_i^T)\ P(C_i^T)\ /P(C_j^I\ )\ P(C_j^I/q)$ represents the probability that $C_i^T$ and q belong to $C_j^I$. This means $P(C_i^T/q)$ is likely larger when q and $C_i^T$ belongs to same category in mediator taxonomy. $P(C_j^I\ )$ shows the size of category $C_j^I$ which guarantees that higher will be the probability that q belongs to $C_j^I$ if higher the probability that q and $C_i^T$ belongs to smaller sized category in mediator taxonomy. This means larger category tends to contain more sub topics while a smaller category contains fewer sub topics. q and $C_j^I$ are related to the same sub topic when they belong to the same smaller category.

## 4.2 WSQTC Processing

When user input any query, it is transformed into a set of pseudo – documents extracted from World Wide Web. Therefore, need for target categories arise. First of all categories are send as search term to a commercial search engine and collect result. For each category, a ranked list of documents which consists of a URL, a title, a snippet of search, URL's webpage content meta keywords and category label is shown. After this a search index is created containing all the documents and web query that need to be classified is issued against this index.

The result of this search includes category selection mechanism for selecting the relevant information. This relevant information is in coordination with target category process through Bridging classifier mechanism and send to bridge for query classification. After this, we applied asynchronous iteration to achieve a great number of simultaneous web searches with lower overheads.

In target taxonomy we want query processor to issue many web searches concurrently without overhead of a parallel query processor. Advance EVScan and RequestSync operators are used for the communication with global RequestPump module. This information from query categorization will be saved in tuple to match exact search.

Figure 3 presents web search query taxonomy classifier flowchart. The steps taken to classify the web queries when bridging classifier is applied and the result is processed concurrently. The query entered for search will move through query categorization to WSQ. In query categorization, with the help of bridge, mediator taxonomy will be coupled with target taxonomy and the result will move for parallel processing. RequestSync operator will synchronize the result and send to Advance EV Scan through dependent join. With the help of Request Pump module for handling the request, the call is placed on a queue when a call is registered with Request Pump and cannot be executed because of limited resources. Then those queued calls are executed.

Figure 4 shows the working of web supported query taxonomy classifier. When queries are entered, they move through query categorization and WSQ to process and get accurate result. In query categorization, the working of query is in offline mode and then online mode. Mediator queries are passed through category selection method. As mediator category is massive, category selection mechanism is used to determine a reduced subset of categories in target category by taxonomy bridging classifier. The target taxonomy and bridging classifier can be associated with query classifier that gives input to WSQ.

The result of query categorization will be saved in WSQ tuples. All the processes will work concurrently. The

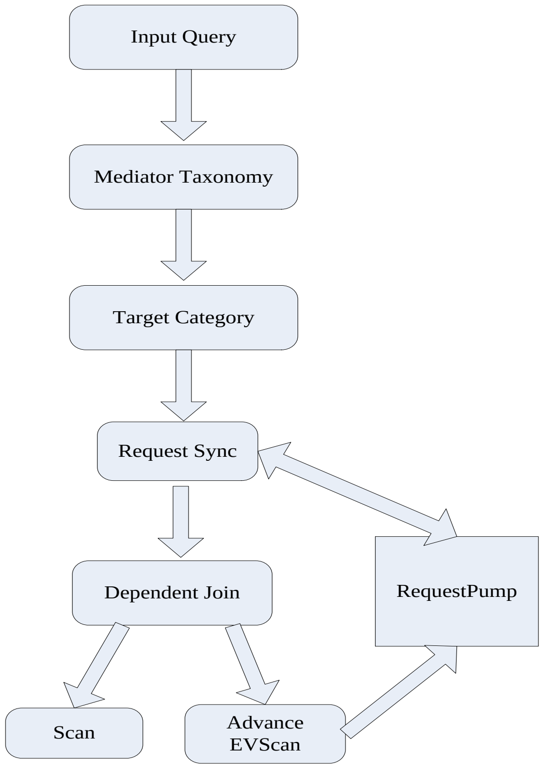


***Figure 3 Web Supported Query Taxonomy Classifier Flowchart***

categories will work through Request Sync with dependent join of Advance EVscan and scan. The sort operator will sort the web-count and dependent join get tuple. Advance EV Scan contact Request Pump and register an external call. Request Pump stores values in Hash table. Request Pump return tuples to RequestSync. The technique used here is general and applies to those situations where concurrency is appropriate for the handling of external source.

## 4.3 Applicability of WSQTC

The broader applicability of web supported query taxonomy classifier is quite general and applies to almost all situations where queries are dependent on values provided by high latency external sources. If a query issues independent calls to many different sources externally or if an external source can handle many parallel requests, this algorithm is suited. It can improve the performance of WSQ queries by a factor of 10 or more.

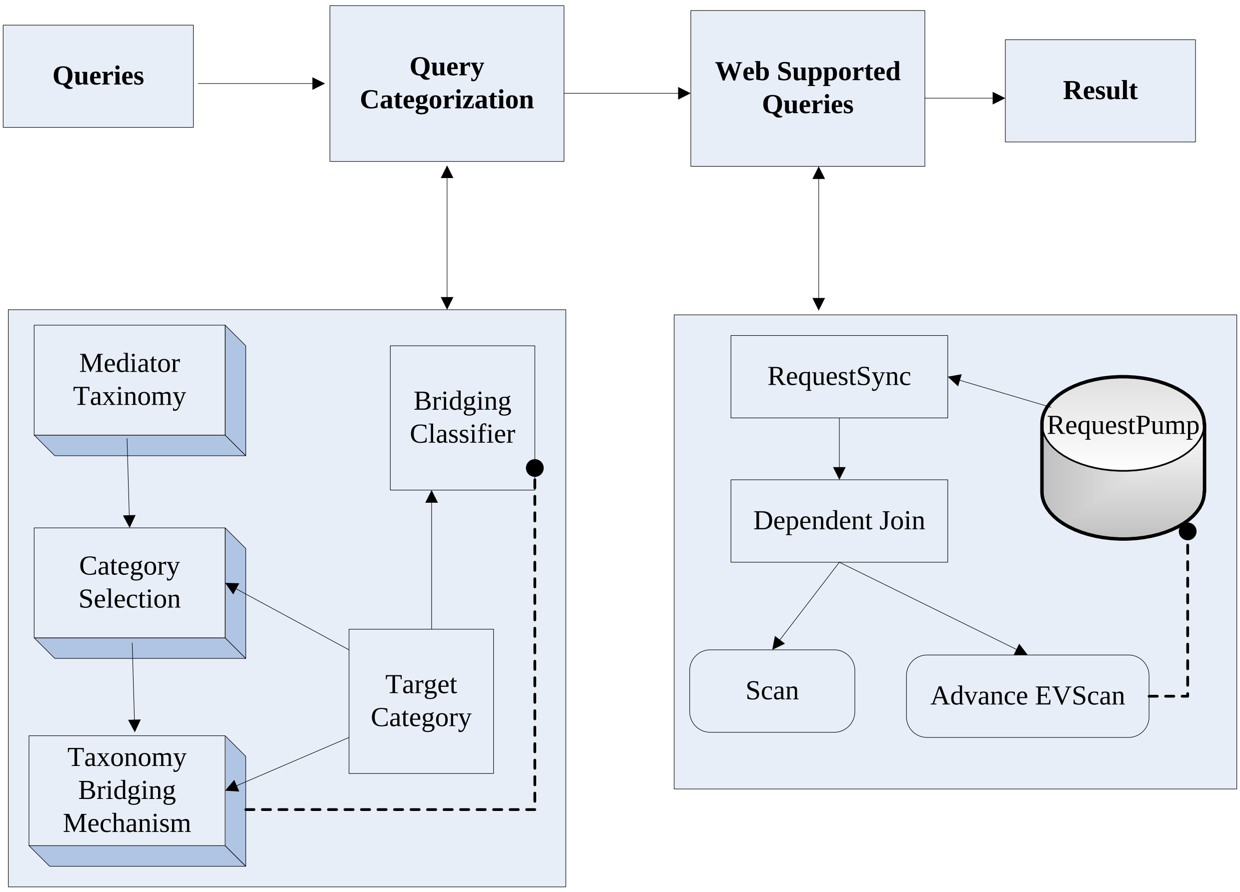


***Figure 4 Web Supported Query Taxonomy Classifier Working***

The speedup may be less impressive for queries involving more complex local query processing over much larger relations. It also provides the solution for classifying web queries into a set of target categories, where the queries are quite short and training data is nonexistent. Bridging classifier are trained from intermediate taxonomy and retraining is not required. Monitoring and controlling resource usages are also solved. Solutions include just altering the dependent join operator to work in parallel, modify the dependent join to branch many threads and prevent concurrency among requests from multiple dependent joins. The query processor block till the first join completes. If external calls overshadow query execution time then asynchronous iteration will provide impressive performance improvements.

## 5. ALGORITHM

The basic steps involved in web supported query taxonomy classifier follows iterator based execution model in which the pattern is dependent on Open, GetNext, Close functions. For detail overview, figure 4 the working of web supported query taxonomy classifier, can be referred.

Step:1 Input queries

Step:2 Mediator Taxonomy work in offline mode and processed against target category.

Step:3 Selection Category method is used to reduce subset of categories in mediator taxonomy relevant to target category.

Step:4 Bridging classifier is associated with Query Classifier that output result with the help of target category.

Step:5 Query Classifier will give this sorted output to WSQ

Step:6 Sort Web-Count

Step:7 RequestSync Synchronize the queries

Step:8 Dependent Join get tuple

Step:9 Advance EV Scan contact Request Pump and register an external call

Step:10 Request Pump stores values in Hash Table

Step:11 Request Pump return tuples to RequestSync

Step:12 Result in form of accurate web searches.

## 6. METHODOLOGY

With the algorithm of Web Supported Query Taxonomy Classifier, the first mapping takes place between input query and an mediator taxonomy. Second plotting is between query from mediator taxonomy and target category. After this the request is send for synchronization. In this way there is no need to create classifier every time when the target organization changes.

The category selection method is used for increasing the usefulness and competence of the queries classified. At the same time, the queries are processed with low overheads and concurrent web searches. This can be achieved by using operators such as Scan and Sort. Dependent join operator is standard nested loop. This operator binds its child nodes with the other one. When it receives a new tuples, it calls Advance EV Scan which is the advance level of virtual table scan that find the data requested.

Advance EV Scan contacts Request Pump and indexed an outer call with tuples. RequestPump module issues asynchronous network requests and responses to each request are stored as they return. If external call is generated, the return data is a value for count. RequestPump stores this value in a Hash table named ReqPump Hash. RequestPump helps Advance EV Scan to indexed it call for achieving concurrency. Now new tuples is generated and the return value is stored in RequestPump.

When the sort operator calls RequestSync, it calls Dependent join below and calls all returned tuples in RequestSync to buffer. In coordination to RequestPump, RequestSync fill in place holders before tuples are returned to its parents. It is RequestPump who generate signals to RequestSync on accessibility of data. RequestSync locates the returned tuples and replace with the count value retrieved from RequestPumpHash. Then it waits for the generation of next signal from RequestPump.

The easiest way to store the tuples precisely and return any complete tuple to its parent besides all incomplete tuples are now made by dependent join. The asynchronous iteration applied here improve query performance by a factor of 10 or more.

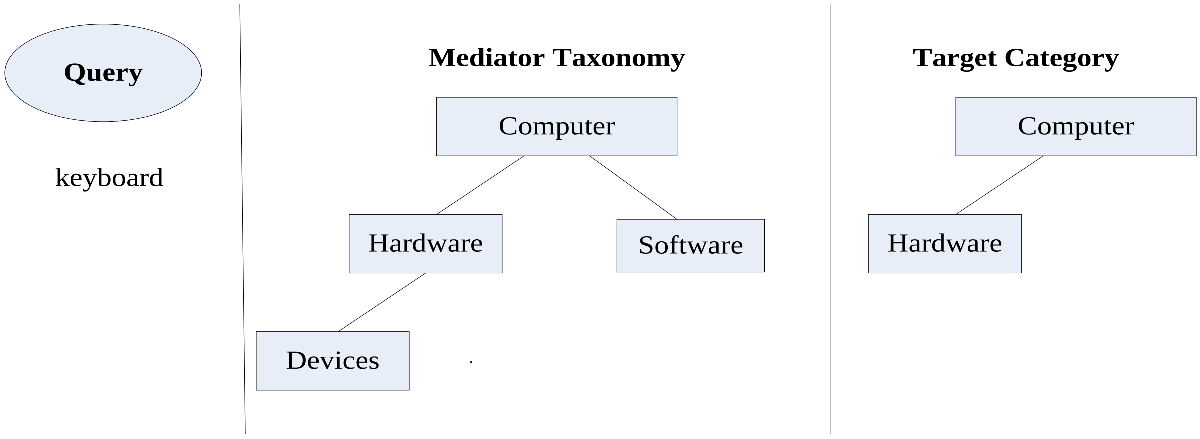


***Figure 5 a Processing of query in the query classification mode.***

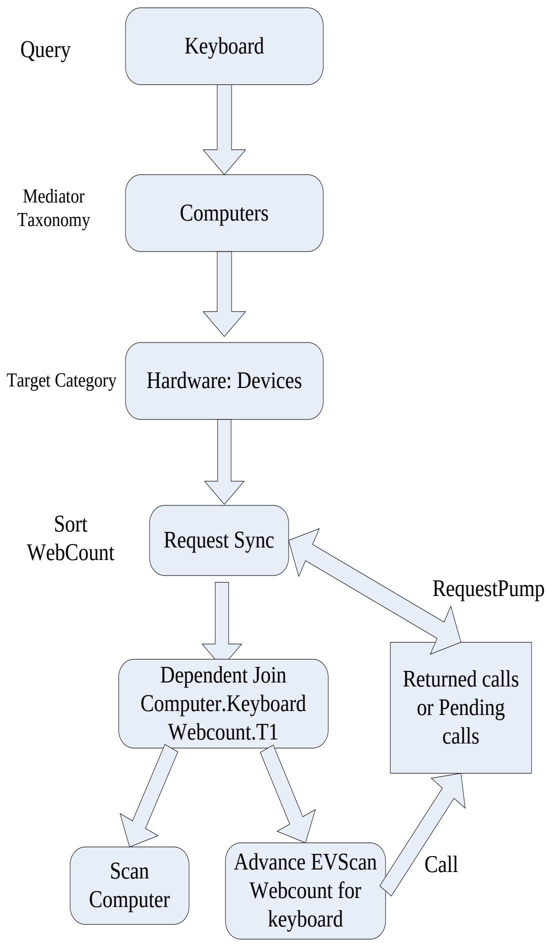


***Figure 5 b Example of searching keyboard query through Web Supported Query Taxonomy Classifier***

## 6.1 Example

Let's discuss one example related to computers. Query is "keyboard". User wants to search about keyboard, a hardware device use for entering into computer. The Mediator taxonomy " Computer \ Hardware \ Device" is directly mapped to Target Category " Computer \ Hardware" as shown in figure 5 a. The tuple will be generated by the table name computer and then the query keyboard is searched. In computer, there are two major categories for devices, hardware and software. According to the request, keyboard will be searched in the hardware category in devices. Webcount rank the table "computer". Sort operator will start ranking the web-count. When dependent join receive that tuples from "computer" it will call Advance EV Scan and Computer.Keyboard will be call. Advance EV Scan also contact RequestPump and indexed the call C with tuples T1 = Computer.Keyboard. C will return count value which will be stored in RequestPump Hash. In order to achieve concurrency and transparency, Advance EV Scan return one web-count tuples T. Dependent Join combines T with Computer.Keyboard and buffer inside Request Sync. The final search result by different search engines will be more synchronized and efficient. In figure 5 b, the complete procedure is shown for the execution of keyboard searching using the method Web Supported Query Taxonomy Classifier.

## 7. CONCLUSION AND FUTURE WORK

This paper concludes that classifying web queries into set of target categories when there is need of connection with mediator taxonomy for improvements in result. Further, sorting is done using asynchronous iteration technique for the usefulness and competency in searching concurrently. Moreover, instead of re-training, the bridging classifier may be rebuilt using mathematical computations any time the target category change. Further described is WSQ approach through which queries are sorted Results from web searches are controlled by WSQ to amplify SQL queries over a relational database. WSQ support those queries which are enabled by two virtual tables whose tuples represent results generated by web search dynamically during query execution. When query processor is idle, Query execution high latency calls are supported by WSQ to one or more than one search engines. In order to enable concurrency between multiple web search requests and query processing, asynchronous iteration technique is formulated. It has broader application and leads to many query optimization problems which is next research topic. In future, we plan to conduct research on the query optimization for mobile servers.

## 8. REFERENCES

[1]. Carlo Meghini, Fabrizio Sebastiani and Umberto Straccia. 2001. A model of multimedia information retrieval. In Journal of the ACM (JACM), Volume 48 Issue 5, September 2001, Pages 909 – 970, New York, NY, USA, ACM.

[2]. Christian Platzer, Clemens Kolbitsch and Manuel Egele. 2011. Removing web spam links from search engine results. In Journal in Computer Virology, Volume 7 Issue 1, February 2011, Pages 51-62, Springer-Verlag New York, Inc. Secaucus, NJ, USA.

[3]. Alessandro Marchetto, Filippo Ricca and Paolo Tonella. 2009. An empirical validation of a web fault taxonomy and its usage for web testing. In Journal of Web Engineering, Volume 8 Issue 4, December 2009, Pages 316-345, Rinton Press, Incorporated.

[4]. Fabrizio Silvestri. 2010. Mining Query Logs: Turning Search Usage Data into Knowledge. In Journal of Foundations and Trends in Information Retrieval, Volume 4 Issue 1—2, January 2010 , Pages 1-174, Hanover, MA, USA.

[5]. Jihie Kim, Peter Will, S. Ri Ling and Bob Neches. 2003. Knowledge-rich catalog services for engineering design. In Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Volume 17 Issue 4, September 2003, Pages 349 – 366, Cambridge University Press New York, NY, USA.

[6]. Ying Li, Zijian Zhen and Honghua Dai. 2005. KDD CUP-2005 Report: Facing a Great Challenge. In SIGKDD Explorations Volume 7.

[7]. Utkasrsh Srivastava, Kamesh Munagala, Jennifer Widom and Rajeev Motwani. 2006. Query Optimization over Web Services. In VLDB'06 September 12-15, 2006, Seol, Korea, ACM.

[8]. Pu-Jeng Cheng, Ching-Hsiang Tsai and Chen-Ming Hung. 2006. Query Taxonomy Generation for Web Search. In CIKM'06, November 5-11, 2006 Arlington, Virginia, USA, ACM.

[9]. Joseph M. Hellerstein, Jeffrey F. Naughton. 1996 Query Execution Techniques for Caching Expensive Methods. In SIGMOD'96 6/96 Montreal, Canada, ACM.

[10]. Evgeniy Gabrilovich, Andrei broder, Marcus Fontoura, Amruta Joshi and Vanja Jasifovski. 2007. Classifying Search Queries Using the Web as a Source of Knowledge. In ACM international Conference on Research and Development in Information Retrieval (SIGIR) Amsterdam, Netherlands.

[11]. S. Chaudhuri,U. Dayal and T. Yan. 1995. Join queries with external text sources: Execution and optimization techbiques. In Proc. of the ACM SIGMOD Intl Conference on Management of Data,San Jose, California.

[12]. Nikos Kirtsis and Sofia Stamou. 2011. Query Reformulation for Task Oriented web searches. In Proc. of IEEE/WIC/ACM Intl conference on Web Intelligence and Intelligent Agent Technology.

[13]. Bettina Fazzinga, Giorgio Gianforme, Georg Gottlob and Thomas Lukasiewicz. 2011. Sementic Web search based on ontological conjunctive queries

[14]. Shuai Ding, Josh Attenberg, Ricardo Baeza and Torsten Suel. 2011. Batch Query Processing for web search engines. In proc. of the fourth ACM intl conference on web search and data mining

[15]. Yuchen Liu, Xiaochuan Ni, Jian Tao sun and Zheng Chen. 2011. Unsupervised transactional query classification based on webpage form understanding. In proc. of 20th ACM intl conference on Information and knowledge management.

[16]. Xianghua Fu, Dongjian Chen, Xueping Guo and Chao Wang. 2011. Query classification based on index association rule expansion. In proc of intl conference on web information systems and mining volume part 2.

[17]. Edwards, J., McCurley, K. S., and Tomlin, J. A. 2001. An adaptive model for optimizing performance of an incremental web crawler. In Proceedings of the Tenth Conference on World Wide Web (Hong Kong: Elsevier Science

[18]. S. Dessloch and N. Mattos. 1997. Integrating SQL databases with content specific search engines. In proc. of 23rd intl conference on very large database, Greece.

[19]. D.Florescu, A. Levy, I. Manolescu and D.Suciu. 1999. Query optimization in the presence of limited access patterns. In proc. of ACM SIGMOD Intl conference on Management of Data, Pennsylvania.

[20]. L. Gravano, V. Hatzivassiloglou and R. Lichtenstein. 2003. Categorizing web queries according to geographical locality. In CIKM 03: Proc of 12th intl conference on Information and knowledge management.

[21]. S.M.Beitzel, E.C. Jensen,O. Frieder, D. Grossman, D. D. Lewis, A. Chowdhury and A. Kolcz. 2005. Automatic web query classification using labeled and unlabeled training data. In SIGIR 05: Proc of 28th annual international ACM SIGIR conference on Research and development in information retrieval.