Search Engine Selection in MetaSearch – A Survey

Sheo Das (Author)
M.Tech Student, AIET Jaipur

Dr. Kuldeep Singh Raghuwanshi(Guide)
Professor CSE Deptt., AIET Jaipur,

Abstract:**Search Engines are widely used for information retrieval, but there are lots of WebPages over the internet and a single search engine cannot cover all the web pages. A Meta search provide the solution for this problem, MetaSearch Engines (MSEs) are tools that help the user to identify relevant information. To perform a Meta Search, user query is sent to multiple search engines; once the search results returned, they are received by the MSE, then merged into a single ranked list and the ranked list is presented to the user. The effectiveness of a metasearch engine is closely related to the search engine selection and result merging algorithm it employs. The algorithm provides the right value information and decision making process to provide necessary data and solve information retrieval problem. In this paper, we focus on the technical challenges of metasearching, namely search engine selection, by providing different algorithms.**

Keywords: Information retrieval, Search engine, Meta Search, Ranking.

1. INTRODUCTION

A person engaged in an information seeking process has one or more goals in mind and uses a search system as a tool to help achieve those goals. Searching relevant information is very difficult due to the explosion of content that has resulted from advances in computer networking, data storage and the availability, type and reliability of information services.

IR is sub field of computer science concerned with presenting relevant information, collected from web information sources to users in response to search. Various types of IR tools have been created, solely to search information on web [19]. Apart from heavily used search engines (SEs) other useful tools are deep-web search portals, web directories and meta-search engines (MSEs) [19]. Search Engines are widely used for information retrieval, two types of search engines exist. General-purpose search engines aim at providing the capability to search all pages on the Web. Special-purpose search engines, on the other hand, focus on documents in confined domains such as documents in an organization or in a specific subject area [1]. But any single search engine cannot solve the problem of Internet information retrieval completely because the search engine has limit to search in their own databases.

Metasearch engines are being developed in response to the increasing availability of conventional search engines [3]. The major benefits of MSEs are their capabilities to combine the coverage of multiple search engines and to reach deep web. Search plans are constrained by the resources available: how much time should be allocated to the query and how much of the Internet resources should be consumed by it [3]. When user poses a query to the Metasearch through the user interface, the Metasearch engine is responsible to identify appropriate underlying search engine which has relevant document with respect to the user query [5]. Meta search engine selects the appropriate underlying search engine with respect to the user query. To enable search engine selection, some information that can represent the contents of the documents of each component search engine needs to be collected first. Such information for a search engine is called the representative of the search engine [17]. To find out the relevant information different similarity measure is used which estimate the relevance between document and user query [5]. The result merger combines all the result into a single ranked list and arranges the documents in descending order with their global similarity with respect to the user query.

The rest of the paper is organized as: In Section 2 Information Retrieval (IR), In Section 3 Web search engine, Section 4 Overview of MetaSearch engine, Section 5 discusses about Search engine selection approach, Section 6 Summary of the work.

2. INFORMATION RETRIEVAL (IR)

Information retrieval deals with techniques for finding relevant (useful) documents for any given query from a collection of documents [1]. The contents of a document may be represented by the words contained in it. In IR system, user passes a query according to their requirement. Query has multiple forms, from which one of the query is passed to the information system. Query may be a Keyword query, Boolean query, Phrase query, Full document query or Proximity query. There are three basic processes an information retrieval system has to support: the representation of the content of the documents, the representation of the user's information need, and the comparison of the two representations [15].

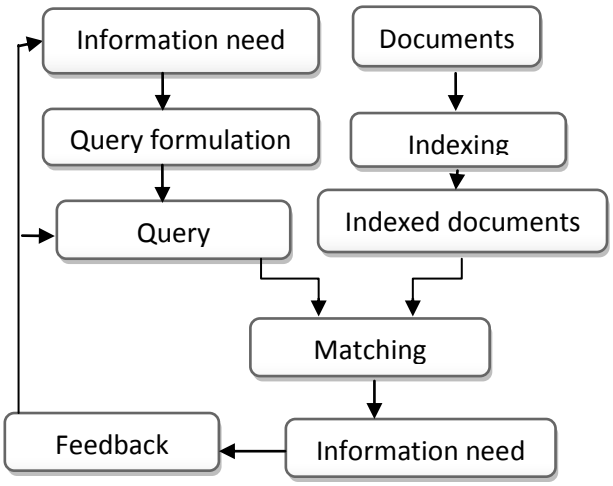

Figure 1: Information retrieval processes

2.1. Information Retrieval Models

There are a large number of hyperlinks in web pages [13] and the mining of so many hyperlinks can bring us lots of useful information, which is great helpful in understanding the semantic of hypertext and providing high quality services to users [13]. It is assumed that hyperlink is the agreement of the web page that the link points to. An IR model governs how a document and a query are represented and how the relevance of a document to a user query is defined. There are three main IR models:

2.1.1 Boolean model

The Boolean model is one of the earliest and simplest information retrieval models. It uses the notion of exact matching to match documents to the user query. Both the query and the retrieval are based on Boolean algebra [20]

In the Boolean model, documents and queries are represented as sets of terms. That is, each term is only considered present or absent in a document. Using the vector representation of the document above, the weight $w_{ij} \in \{0, 1\}$ of term k_i in document d_j is 1 if k_i appears in document d_j, and 0 otherwise [20], i.e.,

$$w_{i,j} = \begin{cases} 1 & \text{if } k_i \text{ appear in document } d_j \\ 0 & \text{otherwise} \end{cases}$$

Boolean Queries: Query terms are combined logically using the Boolean operators **AND**, **OR**, and **NOT**, which have their usual semantics in logic. Thus, a Boolean query has a precise semantics. For instance, the query, ((x AND y) AND (NOT z)) says that a retrieved document must contain both the terms x and y but not z [20].

2.1.2 Vector space model

This model is perhaps the best known and most widely used IR model. Document in the vector space model [7, 20] is represented as a weight vector, in which each component of the index term weight is computed based on some variation of TF and TF-IDF scheme

Term Frequency (TF) Scheme: In this method, the weight of a term t_i in document d_j is the number of times that t_i appears in document d_j, denoted by f_{ij}. Normalization may also be applied. The shortcoming of the TF scheme is that it does not consider the situation where a term appears in many documents of the collection. Such a term may not be discriminative.

Term Frequency-Inverse Term Frequency (TF-IDF) Scheme [20]: Let N is the total number of documents in the system and df_i be the number of documents in which term t_i appears at least once. Let f_{ij} be the raw frequency count of term t_i in document d_j.

Then, the normalized term frequency (denoted by tf_{ij}) of t_i in d_j is given by

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1,j}, f_{2,j}, \ldots, f_{i,j}\}}$$

Where the maximum is computed over all terms those appear in document d_j. If term t_i does not appear in dj then tfij = 0.

The inverse document frequency (denoted by idf_i) of term t_i is given by:

$$idf_i = \log \frac{N}{df_i}$$

The final TF-IDF term weight is given by:

$$w_{ij} = tf_{ij} \times idf_i$$

Query Weight in Vector Space Model

A query q is represented in exactly the same way as a document in the document collection. The term weight w_{iq} of each term t_i in q can also be computed in the same way as in a normal document, or slightly differently. Salton and Buckley [20] suggested the following

$$w_{iq} = \left(0.5 + \frac{0.5 f_{iq}}{\max\{f_{1q}, f_{2q}, \ldots, f_{tq}\}}\right) \times \log \frac{N}{df_i}$$

Document Retrieval and Relevance Ranking: It is often difficult to make a binary decision on whether a document is relevant to a given query. For the vector model, the weight w_{ij} associated with a pair (k_i, d_j) is positive and non-binary. Further, the index term in query are also weighted. Let $w_{i,q}$ be the weight associated with the pair $[k_i, q]$,where $w_{i,q} \geq 0$.Then , query vector q is define $\vec{q} = (w_{1,q}, w_{2q}, \ldots, w_{tq})$ where t is the total number of index in the system. As before, the vector for a document d_j is represented by $\vec{d}_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$.

Therefore, a document dj and user query q are represented as t-dimensional vector as shown in figure. Vector model proposes to evaluate the degree of similarity of document dj with regard to the query q as the correlation between the vector $\vec{d}_j$ and $\vec{q}$ which is the cosine of the angle between these two vectors[9]. i.e.,

$$\text{sim}(dj, q) = \frac{\vec{dj} \bullet \vec{q}}{\left|\vec{dj}\right| \times \left|\vec{q}\right|}$$

$$= \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2 \times \sum_{j=1}^{t} w_{i,q}^2}}$$

Since the $w_{i,j} \geq 0$and $w_{i,q} \geq 0$, sim(q,d_j)varies from 0 to +1. The vector space model ranks the document according to their degree of similarity to the query. Document might be retrieved even if it is partial matching the query in the different document.

2.1.3 Probabilistic models

Several approaches that try to define term weighting more formally are based on probability theory. [7, 15] The notion of the probability of something, for instance the probability of relevance notated as P(R), is usually formalized through the concept of an experiment, where an experiment is the process by which an observation is made. The set of all possible outcomes of the experiment is called the sample space. In the case of P(R) the sample space might be (relevant, irrelevant) and we might define the random variable R to take the values (0, 1) where 0=irrelevant and 1=relevant. [15] Suppose furthermore that $P(D_k)$ is the probability that a document contains the term k with the sample space (0, 1), (0=the document does not contain term k, 1=the document contains term k), then we will use $P(R, D_k)$ to denote the joint probability distribution with outcomes {(0, 0), (0, 1), (1, 0) and (1, 1)}, and we will use $P(R_j \mid D_k)$ to denote the conditional probability distribution with outcomes (0,1). So, $P(R=1 \mid D_k=1)$ is the probability of relevance if we consider documents that contain the term k.

3. OVERVIEW OF WEB SEARCH ENGINE

A Web search engine is essentially an information retrieval system for Web pages. However, Web pages have several features that are not usually associated with documents in traditional IR systems and these features have been explored by search engine developers to improve the retrieval effectiveness of search engines[1]. As a more informed alternative, some of the larger Web search engines attempt to index the Web in its entirety; many smaller Web search engines search considerably more focused databases—names and email addresses or the full text of Shakespeare's plays, for example [3]. Different types of search engines are

Ask is a Search Engine[18], which is also known as Ask Jeeves. It is basically designed to answer the user's queries in the mode of Q&A and is proved to be a focused search engine.

Bing is a Search Engine[18], which was formerly known as Live Search, Windows Live Search, and MSN Search. It is a web search engine (advertised as a "decision engine") that was owned by Microsoft [7].

Google Search or Google Web Search is a web search engine[18] owned by Google Inc. and is the most used search engine on the Web. Google receives several hundred million queries each day through its various services. The main purpose of Google Search is to hunt for text in web pages, as opposed to other data, such as with Google Image Search. Google Search provides at least 22 special features beyond the original word-search capability. These include synonyms, weather forecasts, time zones, stock quotes, maps, earthquake data, movie show times, airports, home listings, and sports scores.

Yahoo! Search is a web search engine[18], owned by Yahoo! Inc. till December 2009, the 2nd largest search engine on the web by query volume, at 6.42%, after its competitor Google at 85.35% and before Baidu at 3.67%, according to Net Applications.

3.1 Challenges faced by Search Engines(SEs)

Using a Search Engine (SE), an index is searched rather than the entire Web. An index is created and maintained by automated web searching by programs commonly known as spiders. Plain search engines prove to be very effective for certain types of search tasks, such as retrieving of a particular URL and transactional queries (where the user is interested in some Web-mediated activity). However, Search Engines can't address informational queries, where the user has information that needs to be satisfied[18].

4. META SEARCH ENGINE (MSE)

A Meta Search Engine overcomes the above by virtue of sending the user's query to a set of search engines, collects the data from them displays only the relevant records[18]. In other words A metasearch engine is a system that provides unified access to multiple existing search engines [1]. Metasearch engine is generally composed of three parts that is, Searching request for pre-processing part, Search interface agent part, Search results processing part [2].

4.1 Metasearch engine components:

A reference software component architecture of a metasearch engine [1, 16] is illustrated in Figure 2.

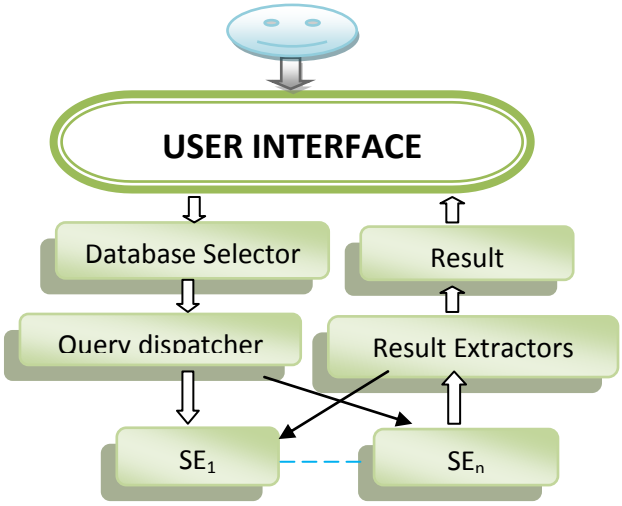

Fig. 2. Metasearch software component architecture.

Database selector: In many cases a large percentage of the local databases will be useless with respect to the query. Sending a query to the search engines of useless databases has several problems. The problem of identifying potentially useful databases to search for a given query is known as the *database selection problem*. The software component *database selector* is responsible for identifying potentially useful databases for each user query.

Document selector: The component *document selector* determines what documents to retrieve from the database of the search engine. The goal is to retrieve as many potentially useful documents from the search engine as possible while minimizing the retrieval of useless documents.

Query dispatcher: The *query dispatcher* is responsible for establishing a connection with the server of each selected search engine and passing the query to it.

Result merger: After the results from selected component search engines are returned to the metasearch engine, the *result merger* combines the results into a single ranked list.

5. SEARCH ENGINE SELECTION

When a metasearch engine receives a query from a user, it invokes the database selector to select component search engines to send the query to. A good database selection algorithm should identify potentially useful databases accurately. Many approaches have been proposed to tackle the database selection problem [6].

5.1 Rough representative approaches:

In these approaches, the contents of a local database are often represented by a few selected key words or paragraphs. Such a representative is only capable of providing a very general idea on what a database is about, and consequently database selection methods using rough database representatives are not very accurate in estimating the true usefulness of databases with respect to a given query.

ALIWEB [1] an often human-generated representative in a fixed format is used to represent the contents of each local database or a site. Note that ALIWEB is not a full-blown metasearch engine as it only allows users to select one database to search at a time and it does not perform result merging.

In **WAIS [1]** for a given query, the descriptions are used to rank component databases according to how similar they are to the query. The user then selects component databases to search for the desired documents. In WAIS, more than one local database can be searched at the same time

5.2 Statistical Representative Approaches

A statistical representative of a database typically takes every term in every document in the database into consideration and keeps one or more pieces of statistical information for each such term.

In D-WISE [1], the representative of a component search engine consists of the document frequency of each term in the component database as well as the number of documents in the database. Therefore, the representative of a database with *n* distinct terms will contain *n* + 1 quantities (the *n* document frequencies and the cardinality of the database) in addition to the *n* terms.

The Collection Retrieval Interface Network (CORI-Net) [1, 5, 12, 14] is carried out using two pieces of information for each distinct term i.e. document frequency and search engine frequency. If a term appears in k document in the search engine, the term is repeated k times in the super document. Super document containing all distinct term in the search engine, As a result, the document frequency of a term in the search engine becomes the term frequency in the super document.

In gGlOSS, the usefulness of a database is sensitive to the similarity threshold used. As a result, gGlOSS [1] can differentiate a database with many moderately similar documents from a database with a few highly similar documents. The computation for estimating the database usefulness in gGlOSS can be carried out efficiently.

Estimating the Number of Potentially Useful Documents (ENPUD) [1]. One database usefulness measure used is "the number of potentially useful documents with respect to a given query in a database." This measure can be very useful for search services that charge a fee for each search. The above methods, while being able to produce accurate estimation, have a large storage overhead. Furthermore, the computation complexity of expanding the generating function is exponential. As a result, they are more suitable for short queries.

Estimating the Similarity of the Most Similar Document.(ESoMSD) [1]: this measure indicates the best that we can expect from a database as no other documents in the database can have higher similarities with the query. On the other hand, for a given query, this measure can be used to rank databases optimally for retrieving the *m* most similar documents across all databases. In this method,

each database is represented by two quantities per term plus the global representative shared by all databases but the computation has linear complexity.

5.3 Learning based approach

These approaches [1] predict the usefulness of a database for new queries based on the retrieval experiences with the database from past queries. The retrieval experiences may be obtained in a number of ways. First, *training queries* can be used and the retrieval knowledge of each component database with respect to these training queries can be obtained in advance (i.e., before the database selector is enabled). This type of approach will be called the *static learning* approach as in such an approach, the retrieval knowledge, once learned, will not be changed. Second, real user queries (in contrast to training queries) can be used and the retrieval knowledge can be accumulated gradually and be updated continuously. This type of approach will be referred to as the *dynamic learning* approach. Third, static learning and dynamic learning can be combined to form a *combined learning* approach. In such an approach, initial knowledge may be obtained from training queries but the knowledge is updated continuously based on real user queries

MRDD Approach. The MRDD (Modeling Relevant Document Distribution) approach [8] is a static learning approach. During learning, a set of training queries is utilized. Each training query is submitted to every component database. From the returned documents from a database for a given query, all relevant documents are identified and a vector reflecting the distribution of the relevant documents is obtained and stored. Specifically, the vector has the format $<r_1, r_2, : : : , r_s>$, where r_i is a positive integer indicating that r_i top ranked documents must be retrieved from the database in order to obtain i relevant documents for the query. With the help of cosine distance similarity function it finds the similarity between user query and all training queries and identifies the k-most similar training query and find the average relevant document distribution vector over k vector corresponding to the k-most similar training queries. Finally, average distribution vector is used to identify the appropriate search engines.

QuerySim

Since there tends to be many similar queries [11] in a real world federated search system, the valuable information of past queries can help us provide better resource selection results. In this section, we propose a novel algorithm, which is called *qSim*, to utilize the valuable information to guide the decision of resource selection. In the algorithm [4, 5] $rel(s_j|q)$, means it is more appropriate search engine contain more relevant information for the user query. The value of $rel(s_j|q)$ depends on $rel(s_j|p_i)$ and $sim(p_i|q_i)$ where $rel(s_j|p_i)$ is the relevance between search engines and past queries and $sim(p_i|q_i)$ is the similarity between all past queries with the user query. The search engines with higher value of $rel(s_j|q)$ being selected by the Metasearch engine.

Savvy Search

Savvy- Search (www.search.com) is a metasearch engine employing the dynamic learning approach. In SavvySearch [1, 3] the ranking score of a component search engine with respect to a query is computed based on the past retrieval experience of using the terms in the query. More specifically, for each search engine, a weight vector (w_1, ….. ,w_m) is maintained by the database selector, where each w_i corresponds to the i^{th} term in the database of the search engine. Initially, all weights are zero. When a query containing term t_i is used to retrieve documents from a component database D, the weight w_i is adjusted according to the retrieval result. If no document is returned by the search engine, the weight is reduced by 1/k, where k is the number of terms in the query.

SavvySearch also tracks the recent performance of each search engine in terms of h, the average number of documents returned for the most recent five queries, and r, the average response time for the most recent five queries sent to the component search engine. If h is below a threshold T_h (the default is 1), then a penalty $P_h = \left(\frac{(T_h - h)}{T_h}\right)^2$ for the search engine is computed. Similarly, if the average response time r is greater than a threshold Tr (the default is 15 seconds), then a penalty $pr = \frac{(r - Tr)^2}{(ro - Tr)^2}$ is computed, where ro = 45 (seconds) is the maximum allowed response time before a timeout. For a new query q with terms t_1,…… , t_k, the ranking score of database D is computed by

$$r(q, D) = \frac{\sum_{i=1}^{k} wt_i \cdot \log(N/f_i)}{\sqrt{\sum_{i=1}^{k} |w_i|}} - (p_h + p_r)$$

where log(*N fi*) is the *inverse database frequency weight* of term *ti* , *N* is the number of databases, and *fi* is the number of databases having a positive weight value for term *ti* .

Profusion

ProFusion approach[5, 21] is a hybrid learning approach, which combines both static and dynamic learning approach. In the ProFusion approach, when a user query is received by the metasearch engine, the query is first mapped to one or more categories. The query is mapped to a category that have at least one term that belong to the user query.

In ProFusion preset categories are utilized in the learning process. The categories are like “Science and Engineering,” “Computer Science”. A set of terms is associated with each category to reflect the topic of the category. For each category, a set of training queries is identified. The reason for using these categories and dedicated training queries is to learn how well each component database will respond to queries in different categories. For a given category C and a given component database D, each associated training query is submitted to D. From the top 10 retrieved documents, relevant documents are identified. Then a score reflecting the performance of D with respect to the query and the category C is computed by [01]

$$C * \frac{\sum_{i=1}^{10} N_i}{10} * \frac{R}{10}$$

Where C is the constant and R is the number of relevant documents among top-10 retrieved documents. Value of N_i is calculated as

$$N_i = 1/i ,$$

if i^{th} ranked document is relevant,
0 otherwise

6. SUMMARY

This paper presented a comprehensive survey and understanding of Meta Search Engines. It is understood that Meta Search Engine exhibits superior performance than any Search Engine. Our survey seems to indicate that better solutions to each of the two main problems, namely Information retrieval and Search engine selection.

REFERENCES:

1. WEIYI MENG , CLEMENT YU , KING-LUP LIU "Building Efficient and Effective Metasearch Engines" ACM Computing Surveys, Vol. 34, No. 1, pp. 48–89, March 2002.
2. XUE YUN, SHEN XIAOPING, CHEN JIANBIN "Research on an Algorithm of Metasearch Engine Based on Personalized Demand of Users" 2010 International Forum on Information Technology and Applications, IEEE, 2010.
3. DANIEL DREILINGER, ADELE E. HOWE " Experiences with Selecting Search Engines Using Metasearch" ACM Transactions on Information Systems, Vol. 15, No. 3, Pages 195–222, July 1997.
4. SULEYMAN CETINTAS, LUO SI, HAO YUAN "Learning from Past Queries for Resource Selection" ACM CIKM'09, November 2–6, 2009.
5. R.Kumar, A.K Giri "Learning Based Approach for Search Engine Selection in Metasearch" IJEMR Volume-3, Issue-5, ISSN No.: 2250-0758, Pages 82-88, October 2013.
6. FILIPPO MENCZER, MELANIA DEGERATU, W. NICK STREET "Efficient and Scalable Pareto Optimization by Evolutionary Local Selection Algorithms" Massachusetts Institute of Technology , Evolutionary Computation 8(2): 223-247, 2000
7. AMIT SINGHAL, "Modern Information Retrieval: A Brief Overview" Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2001.
8. G.TOWELL, E.M. VOORHEES, N.K. GUPTA, B.J LAIRD "Learning Collection Fusion Strategies for Information Retrieval" Appears in Proceedings of the Twelfth AnnualMachine Learning Conference, Lake Tahoe, July 1995.
9. YUAN FU-YONG, WANG JIN-DONG "An Implemented Rank Merging Algorithm for Meta Search Engine" International Conference on Research Challenges in Computer Science, IEEE, 2009.
10. LI QIANG, CHEN QIN, WANG RONG-BO "Weighted-Position & Abstract Ranking Algorithm Based on User Profile for Meta-Search Engine" World Congress on Software Engineering, IEEE, 2009.
11. LUO SI AND JAMIE CALLAN, "Relevant Document Distribution Estimation Method for Resource Selection" SIGIR '03, July 28-Aug 1, 2003, Toronto, Canada. Copyright ACM, 2003.
12. LUO SI and JAMIE CALLAN "A Semi supervised Learning Method to Merge Search Engine Results" ACM Transactions on Information Systems, Vol. 21, No. 4, Pages 457–491. October 2003.
13. LING ZHENG, YANG BO, NING ZHANG "An Improved Link Selection Algorithm for Vertical Search Engine" The 1st International Conference on Information Science and Engineering, Crown Copyright IEEE, 2009.
14. LUO SI AND JAMIE CALLAN "The Effect of Database Size Distribution on Resource Selection Algorithms" SIGIR 2003 Ws Distributed IR, LNCS 2924, pp. 31–42, 2003.© Springer-Verlag Berlin Heidelberg 2003.
15. DJOERD HIEMSTRA "Information Retrieval Models" John Wiley and Sons, Ltd., ISBN-13: 978-0470027622, November 2009.
16. H. JADIDOLESLAMY " Search Result Merging and Ranking Strategies in Meta-Search Engines: A Survey" IJCSI International Journal of Computer Science Issues, ISSN (Online): 1694-0814, Vol. 9, Issue 4, No 3, July 2012.
17. HOSSEIN JADIDOLESLAMY "INTRODUCTION TO METASEARCH ENGINES AND RESULT MERGING STRATEGIES: A SURVEY" International Journal of Advances in Engineering & Technology, ISSN: 2231-1963, Nov 2011.
18. K.SRINIVAS, P.V.S.SRINIVAS, and A.GOVARDHAN "A Survey on the "Performance Evaluation of Various Meta Search Engines" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, ISSN (Online): 1694-0814, May 2011.
19. MANOJ M AND ELIZABETH JACOB "Information retrieval on Internet using meta-search engine: a review" Journal of Scientific & Industrial Research, Vol 67, pp. 739-746 October, 2008.
20. BING LIU "Web DataMining" ACM Computing Classification, 1998.
21. Daniela Rus, Robert Gray, and David Kotz "Autonomous and Adaptive Agents that Gather Information" Proceedings of International Workshop on Intelligent Adaptive Agents, WS-96-04, AAAI '1996