Design of a Metacrawler for Web Document Retrieval

K R Remesh Babu
Assistant Professor
Department of Information Technology
Government Engineering College, Idukki, India
remeshbabu@yahoo.com

A P Arya
Department of Computer Science and Engineering
PSG College of Technology
Coimbatore, India.
aryaanil88@gmail.com

Abstract—**Web Crawlers 'browse' the World Wide Web (WWW) on behalf of search engine, to collect web pages from numerous collections of billions of documents. Metacrawler is similar to that of a meta search engine that combines the top web search results from popular search engines. World Wide Web is growing rapidly. This possesses great challenges to general purpose crawlers. This paper introduces an architectural framework of a Metacrawler. This crawler enables the user to retrieve information that is relevant to the topic from more than one traditional web search engines. The crawler works in such a way that it fetches only the pages that are relevant to the topic. The PageRank algorithm is often used in ranking web pages. But, the ranking causes the problem of topic-drift. So, modified PageRank algorithm is used to rank the retrieved web pages in such a way that it reduces this problem. The clustering method is used to combine the search results so that the user can easily select web pages from the clustered results based upon the requirement. Experimental results show the effectiveness of the Metacrawler.**

Keywords- Search Engine, Web Crawler, Metacrawler, Ranking Algorithms, Clustering.

I. INTRODUCTION

Users' use web search engines to find specific information. As the volume of web pages is increasing rapidly, it is becoming a difficult task to retrieve information that is relevant to the topic.

Web mining technique plays an important role in developing an intelligent web. The useful pages from the web are retrieved by the crawler before applying the web mining techniques. The traditional web crawler traverses the web by following the hyperlinks and stores the downloaded pages in a database.

There are a lot recent researches in new types of crawling techniques such as focused crawling based on semantic web [4], cooperative crawling [5], distributed web crawler [7] and intelligent crawling [8].

The main objective in this paper is to propose a Metacrawler framework. In Section 2, the works related to the web crawler is discussed. Section 3 describes the architectural framework of the Metacrawler. The Sequence Diagram and Activity Diagram are explained in Section 4. In Section 5, the Performance of the proposed System is evaluated. Finally, Section 6 presents the conclusion and future work.

II. RELATED WORK

Agent based web mining systems can be classified into three categories [3]: intelligent search agents, information filtering and personalized web agents. Several intelligent web agents like Harvest [10], ShopBot [11] have been developed to search for relevant information using domain characteristics. Research works based on new types of crawling techniques are as follows:

A new alternative of organizing web documents [4] which emphasizes a direct separation between the syntactic and semantic facets of the web information is proposed. This approach provides a collaborative proximity-based fuzzy clustering and shows how this type of clustering is used to discover a structure of web information by a prudent reliance on the structures in the spaces of semantics and data. The method focuses on the reconciliation between the two separated facets of web information and a combination of results leading to a comprehensive data organization.

A scheme is proposed [5] to permit a crawler to acquire information about the global state of a website before the crawling process takes place. This scheme requires web server cooperation in order to collect and publish information on its content, useful for enabling a crawler to tune its visit strategy. If this information is unavailable or not updated, the crawler acts in the usual manner.

A crawler that employs canonical topic taxonomy [7] to train a naive-Bayesian classifier, which then helps determine the relevancy of crawled pages is proposed. The crawler also relies on the assumption of topical locality to decide which URLs to visit next. Another approach [2] focuses on the classification of links instead of downloaded web pages to determine relevancy. This method combines a Naive Bayes classifier for classification of URLs with a simple URL scoring optimization to improve the system performance.

A crawler with improved PageRank algorithm [3] is proposed. In this improved PageRank algorithm, a web page is

978-1-4673-5118-8 ©2012 IEEE

divided into several blocks by HTML document's structure and the most weight is given to linkages in the block that is most relevant to given topic. The visited outlinks are regarded as a feedback to modify blocks' relevancy. The implementation of this new algorithm helps in resolving the problem of topic-drift. Another improved PageRank algorithm [2] is based on *"topical random surfer"*. The experiment in focused crawler using the T-PageRank has better performance than the Breath-first and PageRank algorithms.

III. METACRAWLER - ARCHITECTURAL FRAMEWORK

The proposed methodology involves the following components: Design of Meta Search Engine, Design of web crawler, Duplicate URL Eliminator, Ranking and Clustering the Results.

A. Design of Meta Search Engine

Web search Engines are designed to search for information from the web. Lots of search engines are currently available. The most popular among them are Google and Yahoo!. All the existing search engines have their own disadvantages. Hence, this has resulted in the creation of numerous search engines.

In the proposed framework, the user input query is given through the main Graphic User Interface (GUI). The search query by the user is processed such a way that all the stop words are removed for better processing of the query. The query is given simultaneously in all the traditional search engines and the results corresponding to the query are extracted from the web database.

The designed meta search engine retrieves major results from three different traditional search engines available. The number of search engines can be selected by the user. The retrieved results are extracted by using the crawler.

The system flow diagram is shown in Figure 1.

B. Web Crawler

Crawler is a means to provide up-to-date data. The web crawler is otherwise called as spiders or robots. A search engine cannot work properly without indexing the web pages. Crawlers are used to create this index.

The crawler visits a web page, reads it and then follows the links to other pages within that page. The crawler returns to the site on a regular basis. Everything that the crawler crawls is indexed. The index is similar to that of a giant book containing the copy of every page that the crawler finds.

Crawler uses the crawling algorithm to crawl the pages from all the search engines. The relevant web pages are retrieved from the web database and are saved.

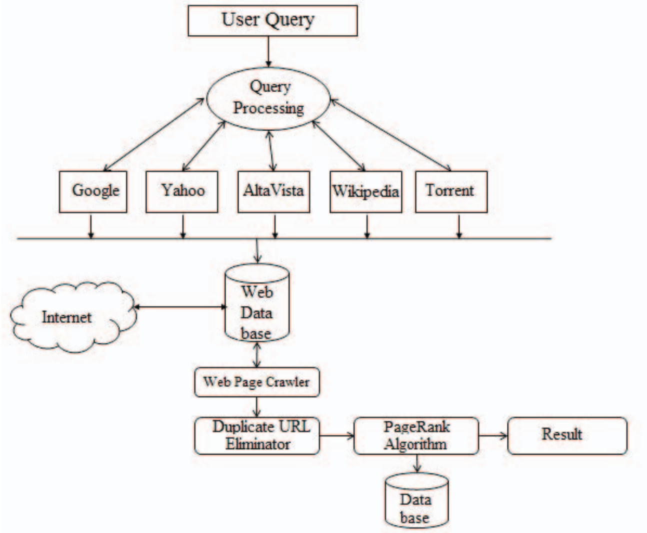


Figure 1: Architectural Framework

C. Duplicate URL Eliminator

The Duplicate URL Eliminator module determines whether an extracted link is already in the URL list or has recently been fetched. Finally, the URL is checked for duplicate elimination. If the URL is already in the list, then it is not added into the list again. And hence, this eliminates the redundant URLs.

D. Ranking

PageRank algorithm is a link analysis algorithm. It is an algorithm for ranking web pages. This algorithm assigns a numerical weighting to each element of a hyperlinked set of documents. The numerical value that this algorithm assigns to any given element E is referred to as the *PageRank* of E.

In general,

$$PR(A) = (1 - d) + d\,(PR(T_1) / C(T_1) + \ldots + PR(T_n) / C(T_n)) \quad \text{-------(1)}$$

Where,

PR(A) is the PageRank of a page A

$PR(T_1)$ is the PageRank of a page T_1

$C(T_1)$ is the number of outgoing links from the page T_1

d is the damping factor in the range of $0 < d < 1$

In the improved PageRank algorithm, a web page is divided into several blocks by HTML document's structure and the most weight is given to linkages in the block that is most relevant to the given topic. The visited outlinks are regarded as feedback to modify blocks' relevancy. The implementation of this new algorithm helps in resolving the problem of topic drift.

The pseudo code of the algorithm is given in Figure 2.

Assume the user interested in topic *k* is browsing a web page *i*. For the next move, the user can either follow an outgoing link on the current page with probability (1-d) or jump to any page uniformly at random with probability d.

```
Starting_urls = searchengine (topic_keyword);
For each link in starting_urls {
    linkscore = sim (topic, url)
    Enqueue (list, link, linkscore); }

While (visited < Maxpages) {
    If (multiplies (visited, frequency)) {
          Recompute score; }
      Link = dequeue_top_link (list);
      Doc = fetch (link);
      Score_sim = sim (topic, doc);
      Enqueue (pages, doc, score_sim);
    If (#pages >= maxpages) {
       Dequeue_links (pages); }
     Merge (list, links(doc), score);
   If (#list > maxpages) {
       Dequeue_links (list); }
}
```

Figure 2: Improved PageRank Algorithm

The *sim*() function returns the cosine similarity between a topic's keywords and a page. The similarity value is calculated by using the following formula.

$$Sim(Q,D) = \frac{\sum_{i=1}^{n} W_{qi} * W_{di}}{\sqrt{\sum_{i=1}^{n} (W_{qi})^2 * \sum_{i=1}^{n} (W_{di})^2}} \qquad \text{------------} \ (2)$$

Where,

w_{qi} is the weight of word w_i in query Q

w_{di} is the weight of word w_i in document D

E. Clustering the Results

Clustering is the act of grouping similar object into sets. One of the most popular clustering techniques is the K-means clustering algorithm. Starting from a random partitioning, the algorithm repeatedly (i) computes the current cluster centers (i.e. the average vector of each cluster in data space) and (ii) reassigns each data item to the cluster whose centre is closest to it. The K-means algorithm terminates when there is no more reassignments take place.

Web users have to shift through the list to locate pages of their interest. This is a time-consuming task when multiple sub-topics of the given query are mixed together. A possible solution to this problem is to cluster search results into different groups and to enable users to identify their required group at a glance.

IV. PROTOTYPE

The section involves the sequence diagram and activity diagram of the system along with the results.

The sequence diagram is shown in Figure. 3.

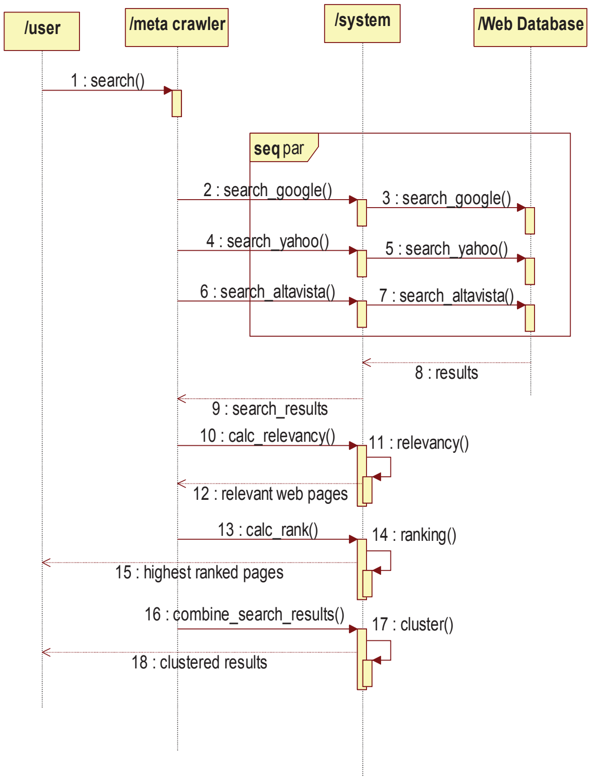


Figure 3: Sequence Diagram

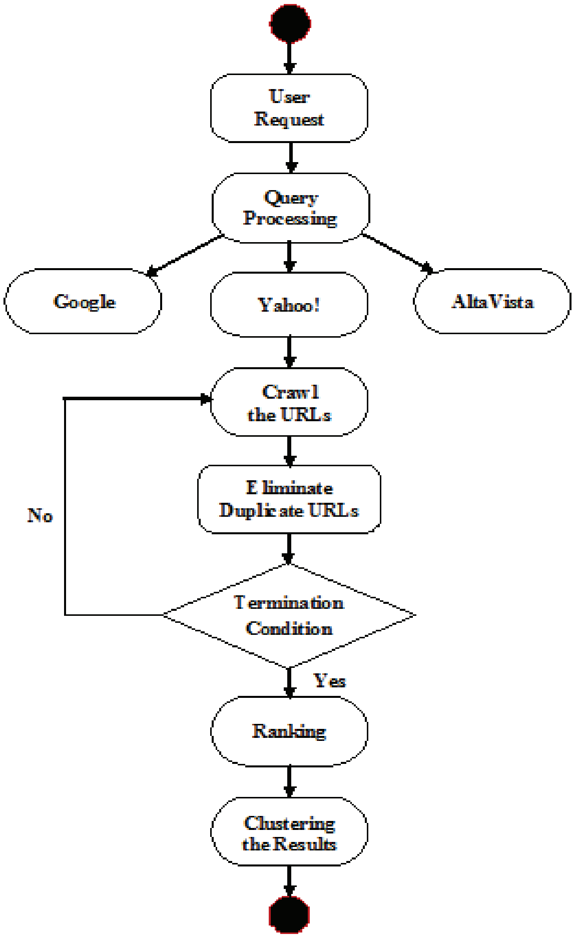


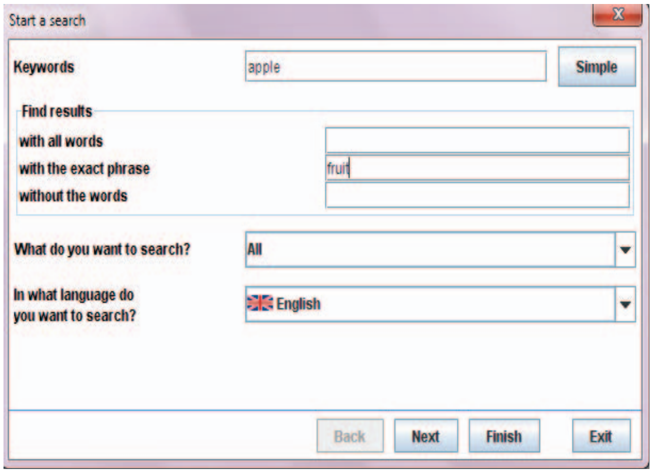
Figure 4: Activity Diagram

User gives the query to be searched to the Metacrawler. The query is executed in parallel in the search engines available. It is represented in the sequence diagram by using the interaction operator ‘par’. The search results obtained are calculated for relevancy and ranking. Pages with high rank are displayed to the user. Finally, the results are clustered into various groups for easy search for the user.

The activity diagram is shown in Figure 4.

Implementation of the Metacrawler along with ranking and clustering of results are obtained.

User can provide the query through the main GUI of the Metacrawler. User can filter the search process by providing the exact keywords to be searched or can add additional details about the words to be searched. The advanced query processing GUI is given in Figure 5.



Keywords: apple

Find results

with all words:

with the exact phrase: fruit

without the words:

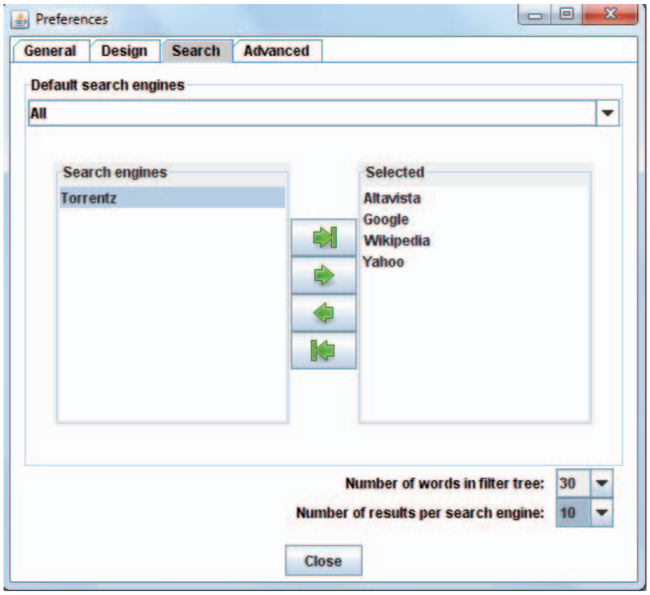
What do you want to search? All

In what language do you want to search? English

Figure 5 Advanced Query Processing

Figure 6 shows the user interface for the user to select any number of search engines from the given list. Also, the user can limit the number of links displayed in a single page using this interface.

After the query is processed, it is given to all the search engines and the search results are retrieved. The relevancy of the pages is calculated based on input query and the web pages are ranked. The pages with highest score are displayed to the user.



Default search engines: All

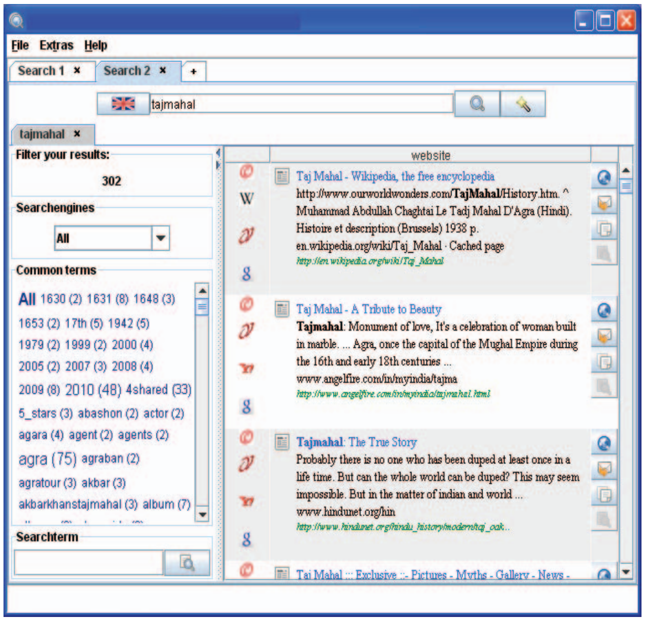
Search engines	Selected
Torrentz	Altavista Google Wikipedia Yahoo

Number of words in filter tree: 30

Number of results per search engine: 10

Figure 6: Search Engine Selector in Metacrawler

The result window for user search for query ‘*Taj Mahal*’ is shown in Figure 7.



tajmahal

302

All

All 1630 (2) 1631 (8) 1648 (3) 1653 (2) 17th (5) 1942 (5) 1979 (2) 1999 (2) 2000 (4) 2005 (2) 2007 (3) 2008 (4) 2009 (8) 2010 (48) 4shared (33) 5_stars (3) abashon (2) actor (2) agara (4) agent (2) agents (2) agra (75) agraban (2) agratour (3) akbar (3) akbarkhanstajmahal (3) album (7)

Taj Mahal - Wikipedia, the free encyclopedia
http://www.ourworldwonders.com/**TajMahal**/History.htm. ^ Muhammad Abdullah Chaghtai Le Tadj Mahal D'Agra (Hindi). Histoire et description (Brussels) 1938 p. en.wikipedia.org/wiki/Taj_Mahal · Cached page
http://en.wikipedia.org/wiki/Taj_Mahal

Taj Mahal - A Tribute to Beauty
Tajmahal: Monument of love, It's a celebration of woman built in marble. ... Agra, once the capital of the Mughal Empire during the 16th and early 18th centuries ... www.angelfire.com/in/myindia/tajma
http://www.angelfire.com/in/myindia/tajmahal.html

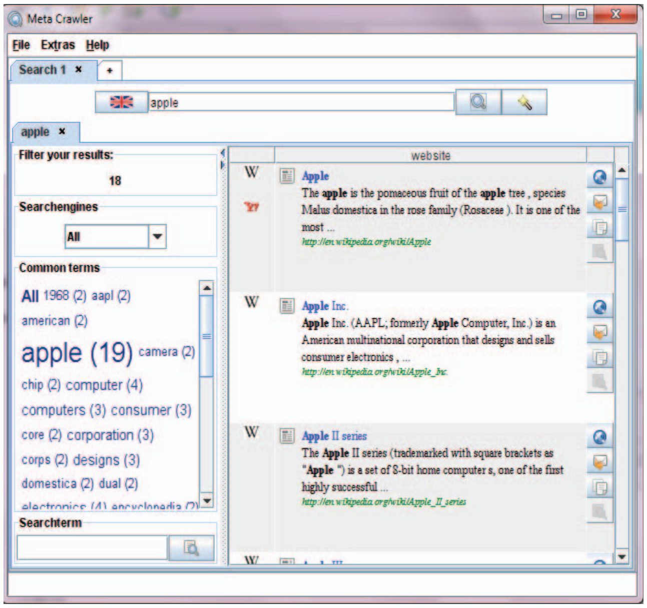
Tajmahal: The True Story
Probably there is no one who has been duped at least once in a life time. But can the whole world can be duped? This may seem impossible. But in the matter of indian and world ... www.hindunet.org/hin
http://www.hindunet.org/hindu_history/modern/taj_oak...

Taj Mahal ::: Exclusive ::- Pictures - Myths - Gallery - News -

Figure 7: Result Window for User Query

The search results are then organized into groups. Since different user has different needs, clustering the results helps the user to search the required query results easily.

Figure 8 shows the result window for clustering of search results for the query '*Apple*' given by the user.



apple

18

All

All 1968 (2) aapl (2) american (2) apple (19) camera (2) chip (2) computer (4) computers (3) consumer (3) core (2) corporation (3) corps (2) designs (3) domestica (2) dual (2) electronics (4) encyclopedia (2)

Apple
The **apple** is the pomaceous fruit of the **apple** tree , species Malus domestica in the rose family (Rosaceae). It is one of the most ...
http://en.wikipedia.org/wiki/Apple

Apple Inc.
Apple Inc. (AAPL; formerly **Apple** Computer, Inc.) is an American multinational corporation that designs and sells consumer electronics , ...
http://en.wikipedia.org/wiki/Apple_Inc

Apple II series
The **Apple** II series (trademarked with square brackets as "**Apple** ") is a set of 8-bit home computer s, one of the first highly successful ...
http://en.wikipedia.org/wiki/Apple_II_series

Figure 8: Clustering Result Window for User Query

V. PERFORMANCE EVALUATION

The performance of the Metacrawler is evaluated with different number of queries. First, the retrieval effectiveness of the crawler is measured using TREC Style Average Precision (TSAP) methodology. The average precision value for five different queries is calculated for all the search engines and is compared with that of the proposed crawler.

Second, the performance of the crawler is evaluated on retrieval relevance ratio.

A. Retrieval Effectiveness using TSAP

The TSAP value for four different search engines for five different queries is calculated and is compared to that of the proposed crawler.

$$\text{TSAP} = \frac{\sum_{i=1}^{N} ri}{N} \qquad \text{-------------} \ (3)$$

Where, N is the number of queries

r_i is the precision for query I

The number of relevant web documents obtained by excuting four different queries on the traditional search engines and the proposed Metacrawler along with average precision value and relevance ratio is tabulated in Table 1.

The Retrieval Effectiveness of the proposed crawler is shown in Figure 9.

Table 1: Number of Relevant Documents

Search Engines	**Yahoo**	**AltaVista**	**Google**	**Metacrawler**
Query (N)				
1	6	6	5	16
2	4	6	7	12
3	5	5	4	12
4	5	4	7	10
Average Precision	0.5	0.52	0.57	0.59
Relevance Ratio	47.6	51.2	57.6	59.2

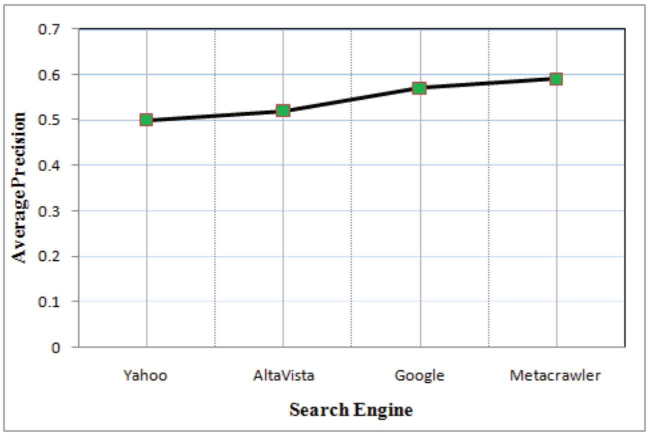


Figure 9: Retrieval Effectiveness

The above graph shows the average precision value calculated using the Equation 3. The results shows that the proposed web crawler has a higher value when compared to three search engines Yahoo, AltaVista and Google. The proposed Metacrawler has given the enhanced performance in terms of retrieval effectiveness and thus it outperforms when compared to the most popular traditional search engines.

B. Relevance Ratio

The relevance ratio of each search engine is calculated using the equation 4 as given below.

$$\text{Relevance Ratio} = \frac{\text{Number of Relevant URLs}}{\text{Total Number of URLs Retrieved}} * 100$$

-------------------- (4)

The graph in Figure 10 shows the relevance ratio of each of the component search engines and the proposed Metacrawler. The graph shows improvement in the relevance ratio by the proposed Metacrawler against the traditional search engines. The proposed Metacrawler has better performance and can enhance the quality of web search results.

The Retrieval Effectiveness using TSAP and Relevance Ratio are used to evaluate the performance of the Metacrawler. The experimental results show that the proposed Metacrawler performs better when compared to other traditional search engines.

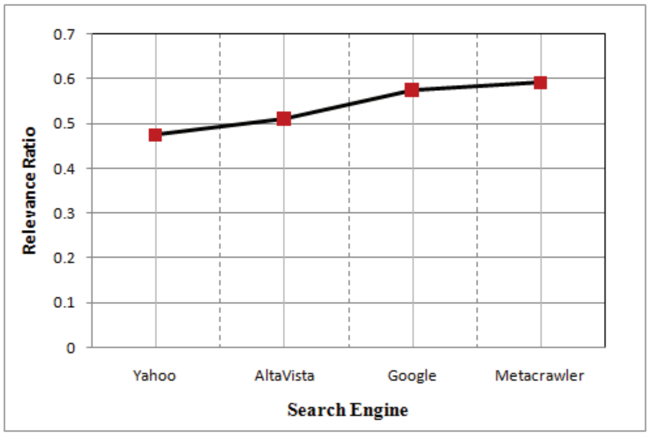


Figure 10: Relevance Ratio

VI. CONCLUSION AND FUTURE WORK

A crawler is similar to that of a program that downloads and stores web pages mainly for a web search engine. An architectural framework of a Metacrawler is proposed based on service oriented architecture. The framework supports selection of participant search engines by the user. Even though, the time taken by the crawler to retrieve the results is a little higher when compared to the other search engines, the proposed crawler works better in terms of its efficiency.

Metacrawler framework can be a good choice for testing new approaches for providing highly relevant search results. Searching techniques can be improved to reduce the search time of the crawler. Crawling algorithms that take structured queries for processing can be implemented, to provide highly relevant personalised search results for every user. Evolutionary algorithms can be used to reduce the search time of the crawler.

REFERENCES

[1] Song Zheng, “Genetic and Ant Algorithms Based Focused Crawler Design”, *Second International Conference on Innovations in Bio-inspired Computing and Applications*, pp. 374-378, 2011.

[2] A. K Elmagarmid, P. G Iperrotis, and V. S Verykios, “Duplicate Record Detection: A Survey”, *IEEE Transactional Knowledge and Data Engineering*, vol.19, No.1, pp. 1-16, January 2011.

[3] Ling Zhang, Zheng Qin, “The Improved PageRank in Web Crawler”, *The 1st International Conference on Information Science and Engineering (ICISE2009),* pp. 1889-1892, 2009.

[4] V. Loia, W. Pedrycz, and S. Senatore. “Semantic Web Content Analysis: A Study in Proximity-Based Collaborative Clustering”, *Proceedings of the 18th International Conference on Data Engineering (ICDE’02).*

[5] V. Shkapenyuk, T. Suel, “Design and Implementation of a High-Performance Distributed Web Crawler”, *International World Wide Web Web Conference*, 2001.

[6] C. C. Aggarwal, F. Al-Garawi, and P. S. Yu, “Intelligent Crawling on the World Wide Web with Arbitrary Predicates”, *Proceedings of the 10th International World Wide Web Conference,* May 2001.

[7] S. Altingovde and O. Ulusoy, “Exploiting Interclass Rules for Focused Crawling”, *IEEE Intelligent System*, 2004.

[8] M. Buzzi. “Cooperative Crawling”, *Proceedings of the First Latin American Web Congress (LA- WEB 2003),* 2003.

[9] S. K. Pal, V. Talwar, “Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions”, *Proceedings of IEEE Transactions on Neural Networks*, Vol. 13, No. 5, 2002.

[10] J. Srivastava, B. Mobasher, R. Cooley, “Web Mining: Information and Pattern Discovery on the World Wide Web”, *International Conference on Tools with Artificial Intelligence*, pp. 558-567, Newport Beach, 1997.

[11] C. M. Bowman, P.B. Danzing, U. Manber, M.F. Schwartz, “Scalable Internet Resource Discovery: Research Problems and Approaches”, *Communications of the ACM*. 37(8), pp. 98-107, 1994.

[12] S. Chakrabarthi, B. Dom, S. Ravikumar, P. Aghavan, S. Rajagopalan, A. Tomkins. “Mining Webs’ Link Structure”, *IEEE Computer*, pp. 60-67, 1997.

[13] F. Gasperetti, A. Micarelli, “Swarm Intelligence: Agents for Adaptive Web Search”, Technical Report, Department of Information, University of ROMA, Rome, Italy, 2000.

[14] Y. Xie, D. Mundlura, V. V. Raghavan, “Incorporating Agent Based Neural Network Model for Adaptive Meta Search”, The Center for Advanced Computer Studies, University of Louisiana, 2004.

[15] S. Raghavan, H. G. Molina, “Crawling the Hidden Web”, *Proceedings of the 27th VLDB Conference*, 2001.

[16] L. Barbosa, J. Freire, “Searching for Hidden-Web databases”, *Eighth International Workshop on the Web and Databases*, 2005.

[17] L. Barbosa, J. Freire, “An Adaptive Crawler for Locating Hidden-Web Entry Points”, *Proceedings of International WWW Conference*, pp. 441-450, 2007.

[18] Animesh Tripathy, Prashanta K Patra, “A Web Mining Architectural Model of Distributed Crawler for Internet Searches Using PageRank Algorithm”, *IEEE Asia-Pacific Services Computing Conference,* pp. 513-518.