A Vertical Search Engine – Based On Domain Classifier

Rajashree Shettar rajshri.shettar@gmail.com
Department of Computer Science,
R.V. College of Engineering,
Bangalore, 560059, Karnataka, India

Rahul Bhuptani rahul.bhuptani@gmail.com
Department of Computer Science,
R.V. College of Engineering,
Bangalore, 560059, Karnataka, India

Abstract

The World Wide Web is growing exponentially and the dynamic, unstructured nature of the web makes it difficult to locate useful resources. Web Search engines such as Google and Alta Vista provide huge amount of information many of which might not be relevant to the users query. In this paper, we build a vertical search engine which takes a seed URL and classifies the URLs crawled based on the page's content as belonging to Medical or Finance domains. The filter component of the vertical search engine classifies the web pages downloaded by the crawler into appropriate domains. The web pages crawled is checked for relevance based on the domain chosen and indexed. External users query the database with keywords to *search;* The Domain classifiers classify the URLs into relevant domain and are presented in descending order according to the rank number. This paper focuses on two issues – page relevance to a particular domain and page contents for the search keywords to improve the quality of URLs to be listed thereby avoiding irrelevant or low-quality ones.

Keywords: — domain classifier**,** inverted index, page rank, relevance, vertical search.

1. INTRODUCTION

The term "search engine" refers to a software program that searches the Web and returns a list of documents in which the keywords are found. Vertical search engines, or domain-specific search engines also called "Vortals", facilitate more accurate, relevant and faster search by indexing in specific domains. Some of the examples of vertical search engines are Financial Search Engines, Law Search Engines, etc. The number of index able web pages is of the order of billions and because of the enormous size of the Web, general-purpose search engines such as Google and Yahoo can no longer satisfy the needs of most users searching for specific information on a given topic. The Broad-Based search engines have gotten broader, so have their search results. This has become increasingly frustrating to users who have turned to search engines to find information on a specialized topic, be it local information, travel sites or specific business channels. The search engine technology had to scale up dramatically in order to keep up with the growing amount of information available on the web [1]. The number of index able web pages is in the order of billions. In contrast with large-scale engines such as Google [2], a search engine

with a specialized index is more appropriate to services catering for specific topic and target groups because it has more structure content and offers more precision. A user visiting a vertical search engine may have a prior knowledge of the domain, so extra input to disambiguate the query might not be needed [3]. Many vertical search engines, or domain-specific search engines, have been built to facilitate more efficient search in various domains. **LookSmart**, an online media and technology company that has launched more than 180 vertical search sites, contends that Web users will increasingly use the Internet the way they do cable television, opting for specialized channels that speak directly to their concerns. In [9] on-line solutions to medical information discovery are presented which tackles medical information research with specialized cooperative retrieval agents. In [10] the document index keeps information about each URL page. It is a fixed width (ISAM), Index Sequential Access Mode index, ordered by document ID. In [11] the Vertical search engines solve part of the problem by keeping indexes only in specific domains. They also offer more opportunity to apply domain knowledge in the spider applications that collect content for their databases. Here the authors use three approaches to investigate algorithms for improving the performance of vertical search engine spiders: a breadth-first graph-traversal algorithm with no heuristics to refine the search process, a best-first traversal algorithm that uses a hyperlink-analysis heuristic, and a spreading-activation algorithm based on modeling the Web. Topic focused crawler [12] is used to collect data. A novel score function is used to evaluate the URL's correlation about the specific topic. Only URLs those whose score is greater than a given threshold is fetched. Factors that contribute to the score are content of the web pages, including the keywords, text and description; the anchor text of the URL; link structure of the URL and pages. In [13] authors use page ranking as a fundamental step towards the construction of effective search engines for both generic (horizontal) and focused (vertical) search. Ranking schemes for horizontal search like the PageRank algorithm used by Google operate on the topology of the graph, regardless of the page content. On the other hand, the recent development of vertical portals (*vortals*) makes it useful to adopt scoring systems focused on the topic and taking the page content into account.

In this paper we propose and present an efficient search engine which takes a seed URL as input. The web pages are crawled based on the domain the URL belongs to i.e either medical or financial domain. The crawler applies indexing techniques for web page analysis and keyword extraction to help determine whether the page content is relevant to a target domain (medical or financial) thereby finding the number of good URLs. Further, domain knowledge is incorporated into the analysis to improve the results, precision rate. The words on the web page are checked against a list of domain-specific terminology and a higher weight is assigned to pages that contain words from the list. Finally, the experimental results are given to assess the features of the relevance score along with the ranked URL for the search keywords provided by the users.

2. WORKING OF VERTICAL SEARCH ENGINE

A vertical search engine searches for specific medical or finance related terms from the crawled web pages. The vertical search engine maintains two lexicons based one for medical and the other for finance domain. The lexicons are built with the knowledge of the domain experts. It evaluates relevance of web pages in context of the domain using content analysis technique. Filter out pages which are not relevant to domain by using TFIDF scores (Term Frequency – Inverse Document Frequency). It is a weighting method used to describe documents in Information Retrieval (IR) problems. The word frequency of the document is calculated. The more a word appears in a document, its term frequency (TF) is high and is estimated to be significant in the document. Inverse document frequency (IDF) measures how infrequent the word is in the document. There are many variants of TFIDF [5] when a user enters a web search query into a search engine (typically by using keyword), the engine examines its inverted index and provides a listing of best matching web pages according to the criteria. The usefulness of the vertical search engine depends on the relevance (information retrieval) of the "result set" it gives back.

3. SYSTEM ARCHITECTURE

The vertical search engine based on domain classifier is built on seven modules; a crawler (spider), HTML parser, filter, domain classifier, page ranker, URLdb, search - supported by a user interface. These modules are described from section 3.1 to section 3.7. Figure 1 depicts the modules involved in building the vertical search engine.

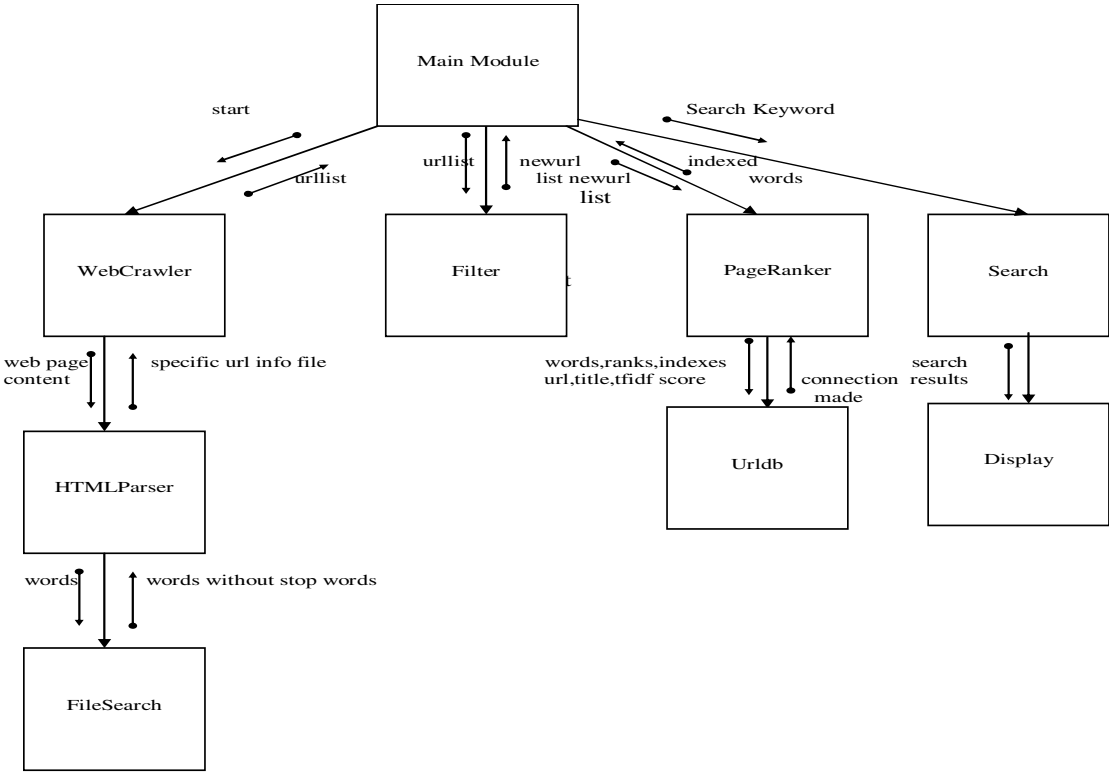


Figure 1: Vertical Search Engine Architecture.

3.1 Web Crawler

The web crawler crawls in a breadth first manner from a given seed URL downloads its contents and retrieves the embedded “Links” and puts them into a queue. The Crawler then recursively takes the URL from the head of the queue and repeats the above procedure till a depth of five or till the queue is empty and the crawling process does not cover the whole web. Crawler handles malformed URLs and robots.txt file. Only HTML pages are crawled and it can handle only HTTP protocol. The crawler calls the HTML parser module to retrieve relevant information from the web page. It also retrieves the last modified time of the page. The working of the web crawler is as shown in figure 2.

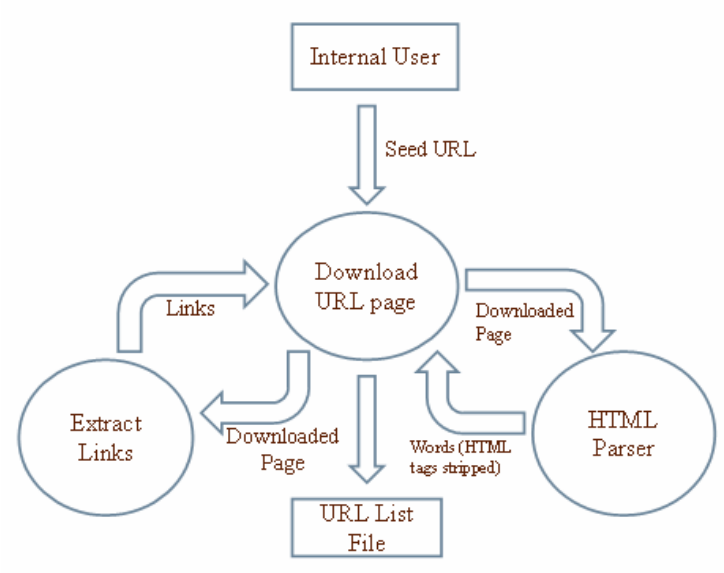


Figure 2: Working of a Web Crawler.

3.2 HTML Parser

HTML parser is the module which tokenizes contents of the file and recognizes the html tags and words. The html tags, style sheets, the stop words (words like to, in and etc which occur frequently and are not content dependant) are stripped off. HTML parser returns the title of the web page along with the words extracted (multiple occurrence taken care) with respective indexes in the page. The HTML parser calls the URL Filter module. An example of the HTML parser output is as given below.

Example: <html><title>hello world</title><body>Hello World! This is our demo for words extracting. </body></html>. The output will be: title: "hello world" |words:Hello(1,3);World(2,4);demo(5);words(6);extracting(7);

3.3 Filter

The search engine being vertical and not generic, all web pages crawled by the spider module from the internet will not be necessary for further processing. Only those web pages which are relevant to the medical domain or finance domain, which contains about 20,000 words relevant to medical field or financial field are selected. Thus it filters the URLs which are queued up by the crawler module into appropriate domains. Hence we need to check how important the page is, and how well it adheres to the topic concerned and discard the URLs which divert from the topic; by analyzing the TDIDF scores. We check each word in a URL page if it exists in the domain lexicon (Medterms.txt/Financeterms.txt). If it does then we calculate its term frequency (TF) and inverse document frequency (IDF) score. The rank of each URL page is the sum of TFIDF scores of all the words in the web page which are also in the domain lexicon. If the sum exceeds a threshold value then we consider those URLs as relevant and index the page. The threshold value is chosen to be the average of the TFIDF scores of the filtered web pages.

$$TFIDF = TF\ (term\ frequency) * IDF\ (inverse\ document\ frequency). \quad (1)$$

$$TF = \frac{(no.\ of\ times\ each\ lexicon\ word\ in\ the\ page\ occurs)}{(total\ no.\ of\ words\ found\ in\ that\ page).} \quad (2)$$

$$IDF = \frac{(total\ no.\ of\ web\ pages)}{(no.\ of\ web\ pages\ in\ which\ the\ lexicon\ word\ occur)} \quad (3)$$

The tokenized words are placed in an inverted index corresponding to medical and finance domains which are created for searching. Inverted index consist of word, URL in which the word is found, rank of the word in this URL.

3.4 Domain Classifier

The filter classifies the URLs crawled based on the two domain lexicons medical and finance. The domain classifier displays the table containing URLs with the TFIDF scores calculated for each domain.

3.5 Page ranker

The success of search engines tends to be attributed to their ability to rank the results. This is non-trivial as the average number of words in a user query is around two; hence the corresponding matching pages tend to be large. The relevant URLs are taken up by the page ranker which ranks them according to their prominence and frequency. Prominence is based on the location where the word occurs (URL itself, title, first paragraph, rest of the body and domain lexicon), thus respective ranks or weights are assigned to the words depending on where they occur in the document. This weight decreases as the location of the word in the document loses its importance. The strategy used for ranking [4] is as shown in the table 1. Example: "intranet" occurs once in URL, once in title (position 2), twice in rest of the body. Rank of "intranet" = (0.50 + 0.60 + 2*0.03) = 1.16.

Word in "URL":	0.50
Word in "title"	1st word : 0.65 2nd word : 0.60 3rd word : 0.55 default : 0.50
Word in "paragraph":	0.25
Word in rest of the page :	0.03 * frequency of the word

Table 1: Page Rank calculation.

If this rank number is *large*, content of the page will be regarded as *fit* to the keyword. The second factor frequency of the word is calculated and added to the prominence weight. This rank calculated for each word is stored along with the URL it occurs in and where it occurs is stored into the database.

3.6 URLdb

This module establishes MySQL database connection and stores the given data into the database. This module stores the URL information –name of the URL, title, first 25 words in the web page. Connection to the database server is established through JDBC interface using MySQL / J Connector com.mySQL.jdbc.Driver. We also specify the location of the database using its URL address: jdbc:mySQL://localhost:3306/mySQL. After successful connection, we create the table "URLinfo" and insert URL details using SQL queries.

3.7 Search

The main purpose of this project is to provide efficient search for the user queries. An intelligent search module has been designed which analyzes the user query and fetches the required results from the database. Figure 3 shows the working of keyword search module. It eliminates stop words in the search query (of, the, and etc). It displays the results in a browser with the title of the web pages along with their links and relevance scores. This module establishes database connection to MySQL and generates a query for retrieval of the web pages according to the keywords supplied by the user through the GUI. It also calculates the new ranks of the web pages (if required) dynamically according to the search.

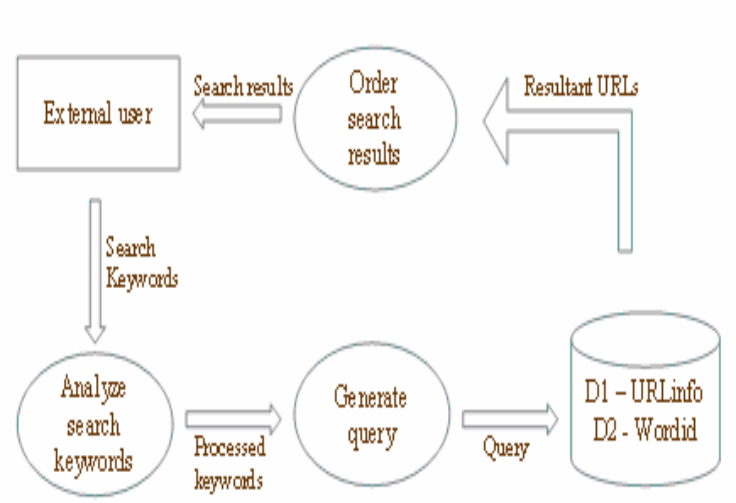
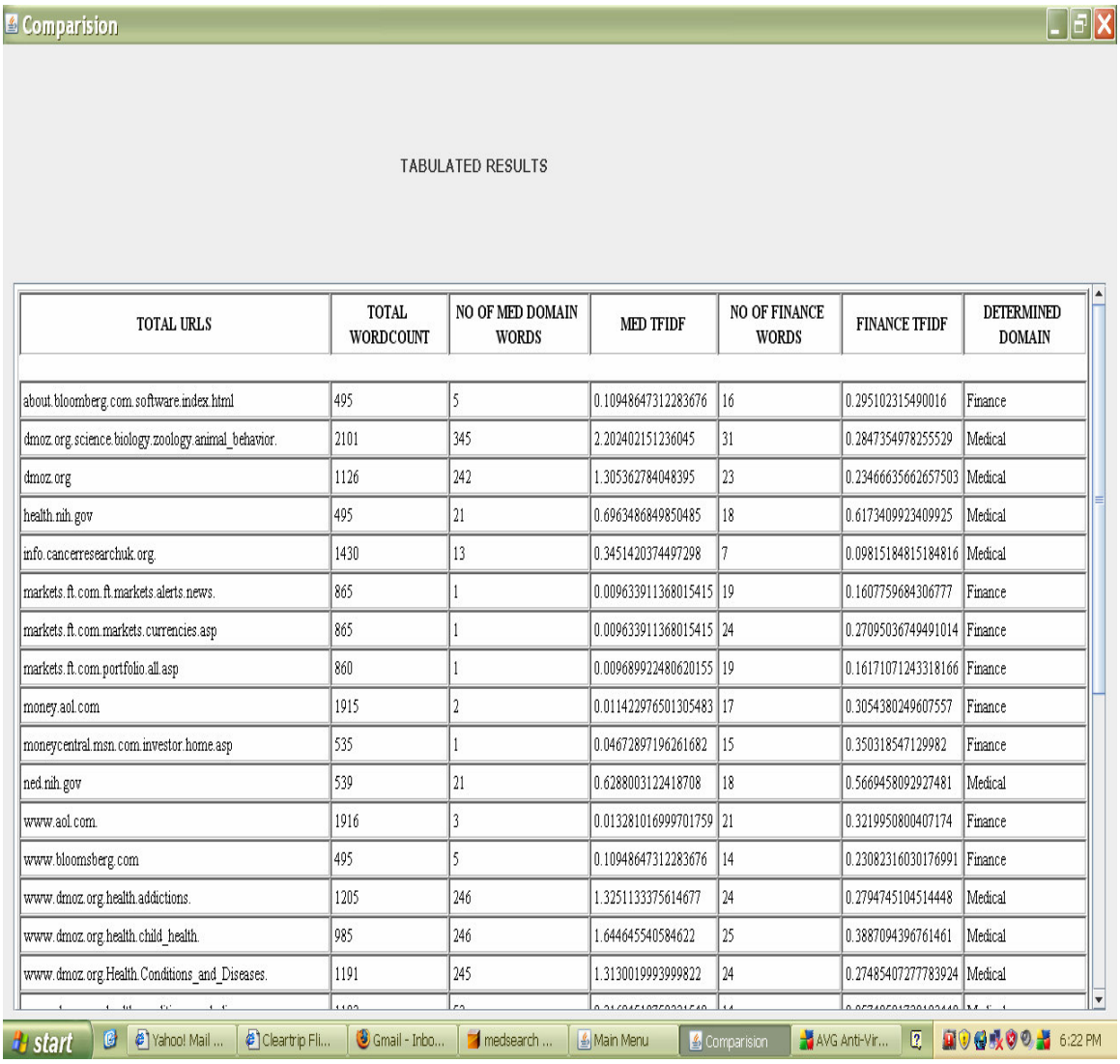


Figure 3: Module for keyword search.

4. EXPERIMENTAL RESULTS

The Vertical search engine is designed to run on limited physical resources. This is developed on Windows XP operating system, using JDK 1.6 and MySQL server 5.0; JDBC APIs for interfacing with the database using mysql-connector-java-5.0.7.; browser used is Internet Explorer. The relevance of each web page to the medical domain and finance domain is evaluated using a measure called TFIDF and a domain lexicon containing about 20,000 words relevant to medical and finance field are used. We calculate the TFIDF value of all words in the web page present in the lexicon. If the sum exceeds the threshold value then we consider it for indexing. The threshold value considered for experiment is the average of the TFIDF scores. The best seed URL (most relevant to the domain) is given to the crawler as input initially and we limit the crawl to of depth 5 from the seed and web pages considered are static. Only keyword search is provided and natural language processing is not provided. Figure 4 shows the list of URLs crawled with total word count and number of medical words and finance words found in the URL webpage along with the TFIDF score. Figure 5 displays the search results on a browser. The URLs listed contain the

search keywords along with their relevance score. The results are listed in a browser so as to enable downloading of the corresponding web pages.
The graph 1 shows total number of URLs crawled and the good URL found after filtering process. Graph 2 is plotted for total URLs crawled and their precision rate. The Graph 3 is plotted for total URLs against the average TFIDF score.



TOTAL URLS	TOTAL WORDCOUNT	NO OF MED DOMAIN WORDS	MED TFIDF	NO OF FINANCE WORDS	FINANCE TFIDF	DETERMINED DOMAIN
about.bloomberg.com.software.index.html	495	5	0.10948647312283676	16	0.295102315490016	Finance
dmoz.org.science.biology.zoology.animal_behavior.	2101	345	2.202402151236045	31	0.2847354978255529	Medical
dmoz.org	1126	242	1.305362784048395	23	0.23466635662657503	Medical
health.nih.gov	495	21	0.6963486849850485	18	0.6173409923409925	Medical
info.cancerresearchuk.org.	1430	13	0.3451420374497298	7	0.09815184815184816	Medical
markets.ft.com.ft.markets.alerts.news.	865	1	0.0096339113680154 15	19	0.1607759684306777	Finance
markets.ft.com.markets.currencies.asp	865	1	0.0096339113680154 15	24	0.27095036749491014	Finance
markets.ft.com.portfolio.all.asp	860	1	0.0096899224806201 55	19	0.16171071243318166	Finance
money.aol.com	1915	2	0.011422976501305483	17	0.3054380249607557	Finance
moneycentral.msn.com.investor.home.asp	535	1	0.04672897196261682	15	0.350318547129982	Finance
ned.nih.gov	539	21	0.6288003122418708	18	0.5669458092927481	Medical
www.aol.com.	1916	3	0.013281016999701759	21	0.3219950800407174	Finance
www.bloomsberg.com	495	5	0.10948647312283676	14	0.23082316030176991	Finance
www.dmoz.org.health.addictions.	1205	246	1.3251133375614677	24	0.2794745104514448	Medical
www.dmoz.org.health.child_health.	985	246	1.644645540584622	25	0.3887094396761461	Medical
www.dmoz.org.Health.Conditions_and_Diseases.	1191	245	1.3130019993999822	24	0.27485407277783924	Medical

Figure 4: User Interface showing the domain classification based on medical and finance lexicons for the list of URLs crawled with total word count and number of medical words and finance words found in the URL webpage along with the TFIDF score.

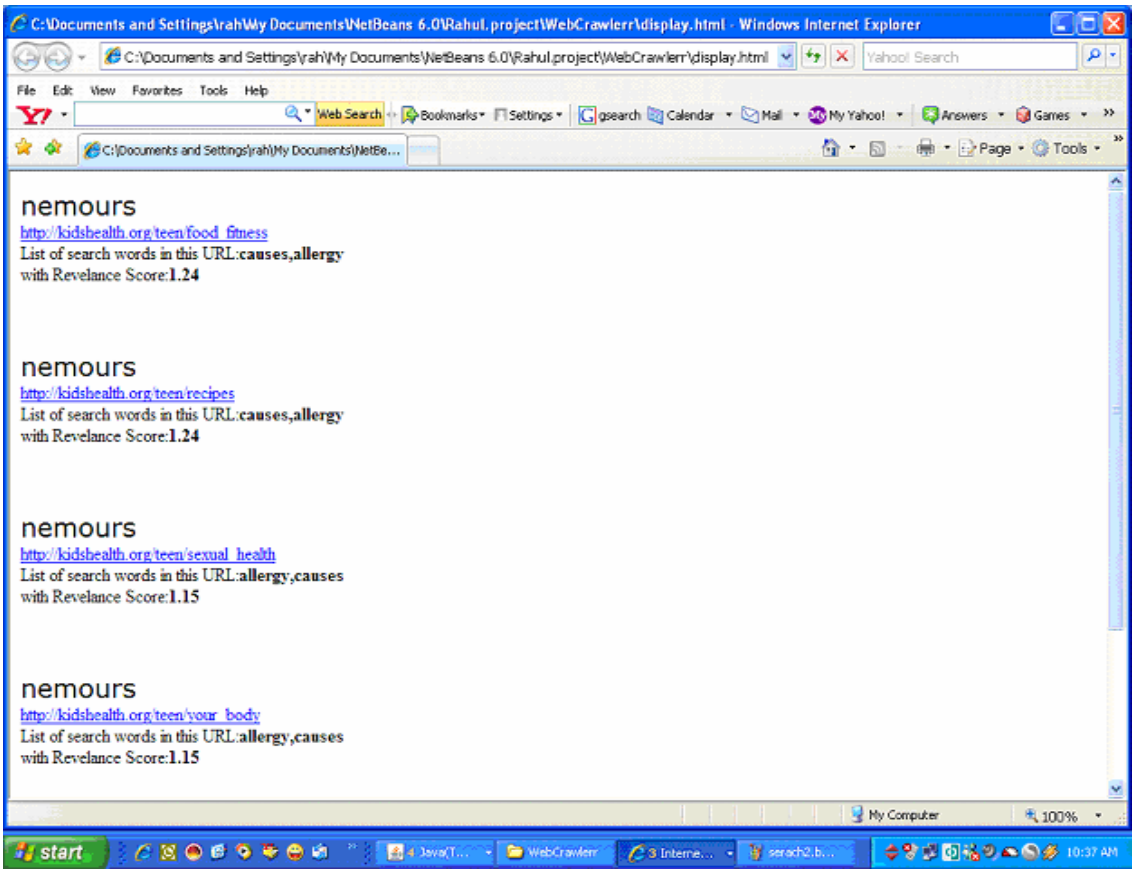
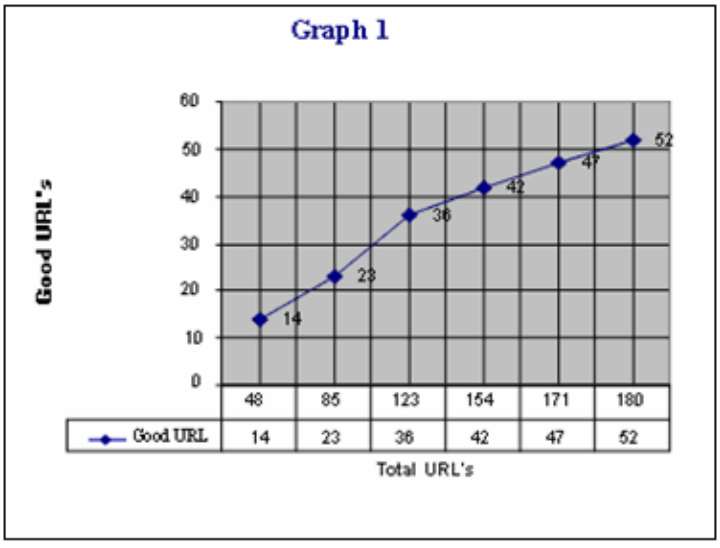


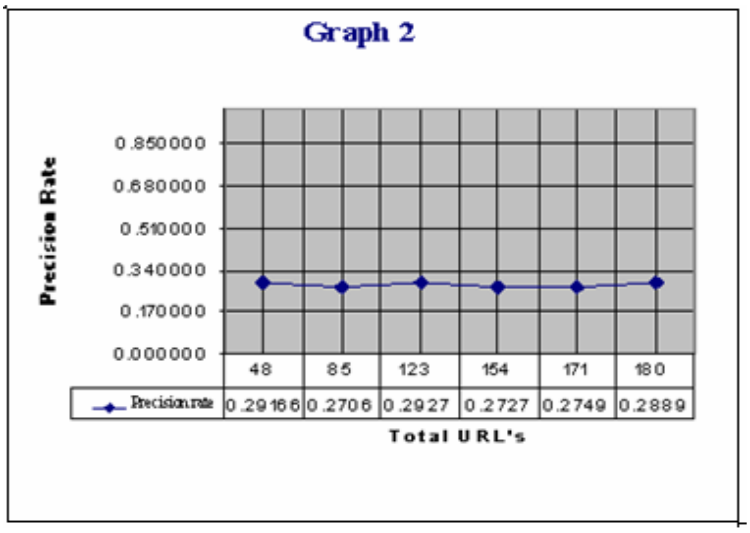
Figure 5: User Interface showing the result of the search operation with the search keywords contained in the web pages of the URLs listed along with their relevance score.



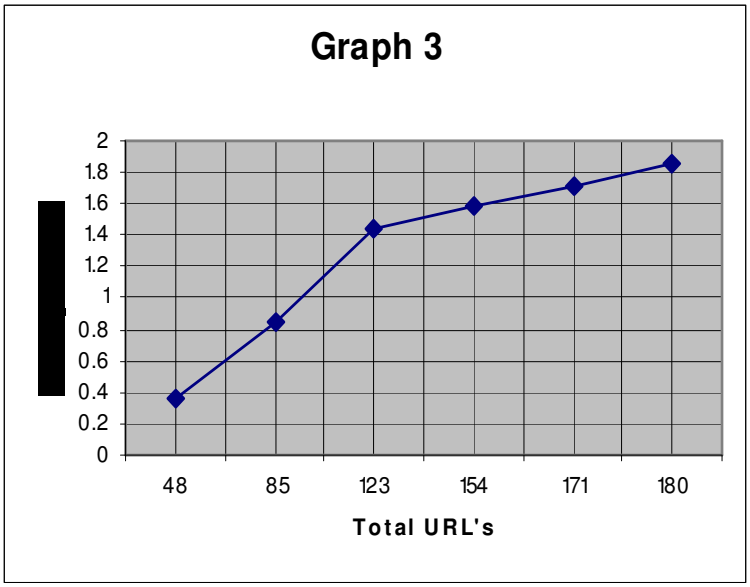
Graph 1: Total number of URL's crawled against the total number of Good URLs extracted by the Vertical Search Engine.

The precision rate is calculated as follows:

$$\text{Precision rate} = \frac{\text{Number of good URLs}}{\text{Total number of crawled URLs}} \quad (4)$$



Graph 2: Total URLs crawled and their precision rate.



Graph 3: Total URLs against the average TFIDF score.

5. CONCLUSION

Vertical search engine is a fast emerging technology, giving serious competitions to generic search engines. The main aim of this paper is to provide users with highly relevant results for medical and finance domain, thus saving user the precious time of avoiding wading through irrelevant search results, hence providing better choice than a generic engine. In this paper we have presented the architecture and implementation details of a vertical search engine. We have indexed about 1, 00,000 words from about 180 URLs. The URLs which are irrelevant i.e. not pertaining to the medical and finance domain are filtered out based on the TFIDF threshold value. Only URL links which contain the search keywords specific to the medical and finance lexicon are listed. The vertical search engine model presented gives efficient results pertaining to the domain chosen. Thereby reducing the users search time for specific topic and giving the user a more relevant and specific URL list for search operation.

6. FUTURE ENHANCEMENTS

The vertical search engine can be enhanced by including phrase search and algorithms like link analysis can be added to provide better results. The domain classifier can be extended for more domains by training the network to automatically classify the domains each web page belongs to using neural network strategy.

7. REFERENCES

[1] George Almpanidis, Constantine Kotropoulos, and Ioannis Pitas. Aristotle University of Thessaloniki, Department of Informatics. "Focused Crawling Using Latent Semantic Indexing" – *An Application for Vertical Search Engines.*

[2]Google Search Technology. Online at http://www.google.com/technology/index.html.

[3] R. Steele, "Techniques for Specialized Search Engines", *in Proc. Internet Computing, Las Vegas, 2001.*

[4] Ng Zhug Whai, "A new city university search engine", Department of information technology.

[5] Pascal Soucy, Guy W. Mineau, Beyond TFIDF weigting for Text Categorization in the Vector Space model, 2005.

[6] Manber, U., Smith, M., and Gopal, B. "WebGlimpse: Combining Browsing and searching", in *Proceedings of the USENIX 1997 Annual Technical Conference.*

[7] Monica Peshave, "How search engine works and a Web Crawler Application", Dept of Computer science,University of Illinios at Springfield, Spingfield,IL 62703.

[8] Castillo, C. (2004). "Effective Web Crawling", *PhD thesis, University of Chile*.

[9] Baujard, O., Baujard, V., Aurel, S., Boyer, C., and Appel, R.D. "Trends in Medical Information Retrieval on the Internet", *Computers in Biology and Medicine, 28,1998.*

[11] Michael Chau and Hsinchun Chen, "Comparison of Three Vertical Search Spiders", *Journal of Computer* ,Vol. 36, No. 5, 2003, ISSN 0018-9162, pp. 56-62, publisher IEEE Computer Society.

[12] Ye Wang, Zhihua Geng, Sheng Huang, Xiaoling Wang, Aoying Zhou, "Academic Web Search Engine – generating a Survey Automatically", Department of ComputerScience, Fudan university, China.

[13] "Web Page Scoring Systems for Horizontal and Vertical Search", Michelangelo Diligenti , Marco Gori,Marco Maggini , Siena, Italy.