STAIR : A System for Topical and Aggregated Information Retrieval

C.V. Krishnakumar, Krishnan Ramanathan

HP Laboratories
HPL-2009-51

Keyword(s):
STAIR, Search, Focused Crawling, Information Retrieval

Abstract:

Web content has exploded dramatically in the last decade and search is becoming increasingly complex. In the current search paradigm, the user has to enter the query and is immediately presented results that are typically accessed sequentially. However, there are scenarios where the above model is not appropriate, either because results being in consumable form is more important than immediacy of results, or because the it is difficult and time consuming to navigate the results in sequential fashion. In this work, we describe the architecture, implementation and utility of STAIR- The System for Topical and Aggregated Information Retrieval, that uses a variant of focused crawling and retrieves relevant information from the web. We present a new interface that selects search results from different search engines, ranks the results and presents the most relevant results as an aggregated PDF document.

External Posting Date: March 6, 2009 [Fulltext] Approved for External Publication
Internal Posting Date: March 6, 2009 [Fulltext]

Published and presented at the First International Conference on HCI, Allahabad, India. Jan 20-23, 2009

STAIR : A System for Topical and Aggregated Information Retrieval

[1]C.V.Krishnakumar[1] Krishnan Ramanathan[2]
[1]Stanford University, California, USA.
[2]HP Laboratories, Bangalore, India

Web content has exploded dramatically in the last decade and search is becoming increasingly complex. In the current search paradigm, the user has to enter the query and is immediately presented results that are typically accessed sequentially. However, there are scenarios where the above model is not appropriate, either because results being in consumable form is more important than immediacy of results, or because the it is difficult and time consuming to navigate the results in sequential fashion. In this work, we describe the architecture, implementation and utility of STAIR- The System for Topical and Aggregated Information Retrieval, that uses a variant of focused crawling and retrieves relevant information from the web. We present a new interface that selects search results from different search engines, ranks the results and presents the most relevant results as an aggregated PDF document.

Keywords : STAIR, Search, Focused Crawling, Information Retrieval

1 Introduction

Search engine technology has had to scale dramatically to keep up with the growth of the web. The one-size-fits-all approach that is being used by the general purpose search engines today is increasingly becoming irrelevant today.Search interfaces today are geared to providing results immediately and getting users to click relevant ads. The results are presented as a sequence of links and snippets from the linked-to document.

The need is for a system that could provide the most "relevant" information about the given query within acceptable time limits.Our solution is to design a information assistant that queries multiple search engines based the information need, selects and consolidates the results and presents them to the user in a compact and consumable manner. The response is provided as a PDF document containing multiple articles. Navigating the consolidated document is much simpler, the user gets more information (compared to search result snippets) that enables her to quickly decide whether to read the content or move to the next result. We believe such an interface would be even better suited for newer kinds of internet access points such as in

- mobile devices, where there is a chance of losing internet connectivity while navigating search results (because the user is on the move and the connection drops) and where it is more cumbersome to surf through multiple results on multiple web pages.

[1] Work performed when CV Krishnakumar was an intern at HP Labs India

- touch based devices where the traditional keyboard/mouse based interaction does not provide as good an experience as our new interface can.

In this paper, we present *STAIR* - System for Topical and Aggregated Information Retrieval that implements focused crawling and retrieves only the relevant information from the web. It compiles, consolidates and processes the information to provide an aggregated PDF document.

2 Related Work

Many different approaches have been proposed by the researchers in the recent years to achieve the goal of improving the efficiency and the accuracy of the search engines, by avoiding the fetching of irrelevant pages from the web. The *focused crawler* introduced by Soumen Chakrabarti[9] used a topical taxonomy and graph distillation to track topical hubs. The Volant system [7] provides a information retrieval paradigm taking post-query navigation into account. White et.al [8] propose a machine learning-based approach for supporting switching search engines by estimating in real time whether more accurate results exist on alternate search engines. The Clusty search engine from Vivisimo clusters results and presents them using the desktop metaphor of folders. Chakrabarti also suggested resource discovery through examples [3]. One other important work in this field has been the combination of link and text analysis for focused crawling by Almpanidis. The hyperlink features for personalization such as URLs, tokens and anchor texts were suggested by Aktas [11]. Perhaps, the work closest to our system is BINGO [2] which provides an architecture for focused crawling. However, the overall empahsis in *BINGO* is on focused crawling unlike our system which places an equal emhasis on all aspects of information retrieval such as fetching web pages, query expansion, focused crawling, ranking and presentation.

3 Motivation and Problem Definition

In recent years, the World Wide Web has grown at a rapid pace [1] and users need ways of quickly finding material on the topic of their interest. In order to achieve this goal, there are a number of search engines that facilitate this goal. However, different search engines have different coverage of the web. Although meta-search engines have been tried in the past, they have been unsuccessful largely because they are unable to scale on the server side. Moreover,there is a tradeoff between getting results immediately and getting relevant results. The user may be willing to wait for some time (e.g. 10 minutes) if the search engine could do a better job of filtering the results. There is no interface for specifying user wait time. Thus, the conventional search engines attempt to serve everyone collect and index all the documents on the web.

The most popular web search engines of today use a *one-size-fits-all* approach in order to serve the entire world. Since the queries are user specific, there is a need for personalizing the web for each user. We need a mechanism whereby we can retrive and store only the most relevant pages from the web, for a specific user. To achieve this end, we use a focused crawling method, so as to retrieve and index only the most relevant pages from the web.In this manner, we also hope to leverage the knowledge of the user specified topics beforehand. Also, the current web search engines, assist in navigating to websites in addition to assisting information retrieval, we have not supported this functionality and restrict ourselves to supporting only information seeking,

Our goal is to create an *aggregated* and *personalized Information Retrieval* (IR) system that **compiles** and **consolidates** the most relevant information on particular topic(s) from the web and automatically creates a *document* providing comprehensive info on topic. The system can be deployed in primarily two possible ways:

The system can be deployed as a desktop-based application wherein, in conjugation with the User profile generator, it would take the input as the user profile and generate a *consolidated document* for each of the topics the user is interested in [Fig 1].

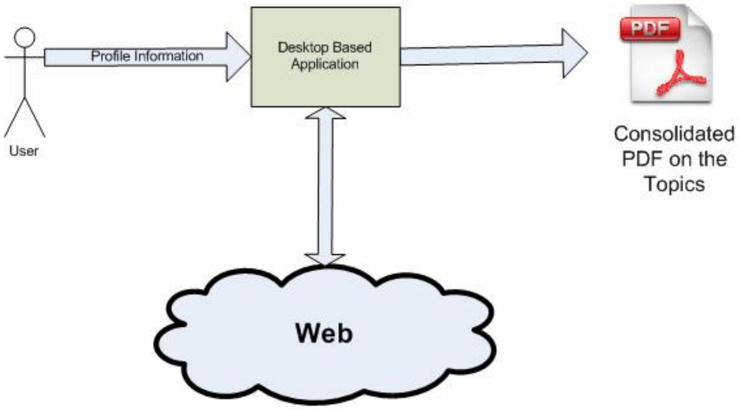


Figure 1: The System as a black box

The system can also be deployed as a *Web Service* wherein, the system would reside on the Server,the user would give his topics of interest and the system would generate the *consolidated document* based on the current relevant information extracted from the web.

4 Architecture and Implementation

The main components of the system are depicted in the Fig. 2. The major components of the system are a crawl seeder, a focused crawler, an indexing mechanism for the crawled pages, a ranking module, the aggregated document generator and the user interface. After the user enters the query, it is presented to search engines (Google, Yahoo) and the DMOZ repository and a set of seed pages (for focused crawling) is retreived. Optionally, if a user profile is available, the query might be formed from topics in the user profile. We also query Delic.io.us for the tags of the pages in the seed set, the similarity between the delicious tags and the query is used to influence the ranking of the page. The list of seed pages form the first set of “frontier” urls for crawling.

For focused crawling, we use a modified version of breadth-first-search known as the best-first-search for selective topic extraction. Depending on the (user-specified) depth of crawl, the links on the retrieved document are also explored. For instance, in a crawl of depth 3, the links on the initially crawled pages are retrieved and then the links on these retrieved pages are crawled and retrieved. The crawl is very lightweight, typically not more than 200 web pages will be fetched, and the crawl can be easily done from a client device. The primary advantage of focused crawling is that the space complexity is minimum since only the most relevant links are fetched and indexed at each iteration. However, this comes at the cost of time complexity since, the system should now combine the evaluation phase with the crawling phase to predict the most rewarding set of links beforehand. This has been implemented by heuristics in the current version of *STAIR* wherein we predict the score of the child link as being times that of the parent node and then consider just 25% of the frontier List at each stage.

In the Analyze phase of the crawl, for each of the fetched pages, we retrieve synonyms for each word from Wordnet and index it alongwith the original word. This is required because pages in DMOZ and Delicious may not have the query words in the document (e.g. a page tagged as “recipe” may not actually have the word “recipe”). The fetched pages are then ranked and selected. The rank of a page is computed as

$$\text{Rank} = k*(\text{TFIDF}) + (1-k)\ (\text{DVCS})$$

TFIDF is the standard IR score, DVCS is the cosine similarity of the query vector with the delicious tags vector. By varying k, we can assign more importance to the presence of the query in the pages or its similarity to the Delicious tags. We have used k=0.5 in our experiments. Pages with higher rank are added to the frontier list for the next iteration of the crawl.

After the crawl terminates, relevant pages are stored in the Lucene index. We query the index and rank the pages according to the equation described above. We then select the top N pages (N is specified by the user).The final step after pages have been selected is to cleanup the HTML, extract sentences and generate a composite PDF document . We performed these steps using off-the-shelf tools.

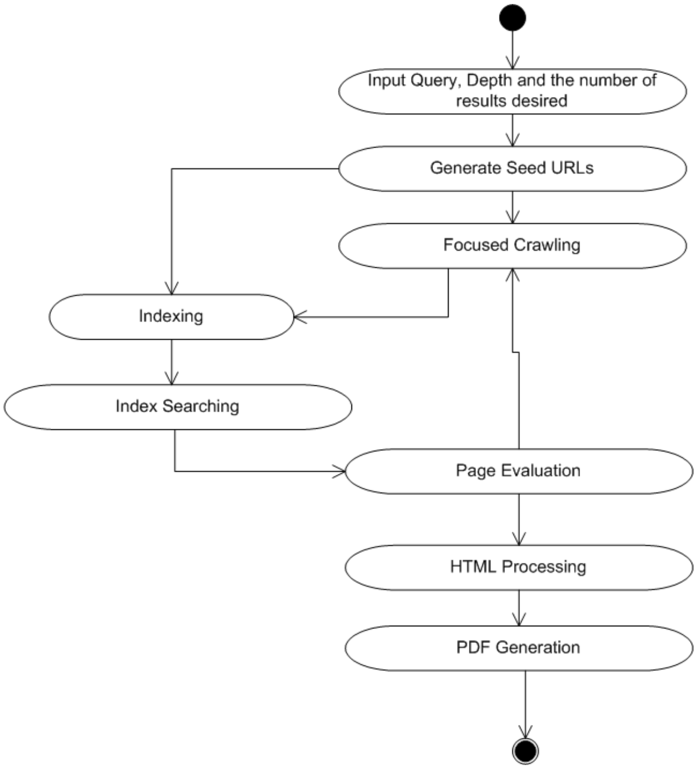
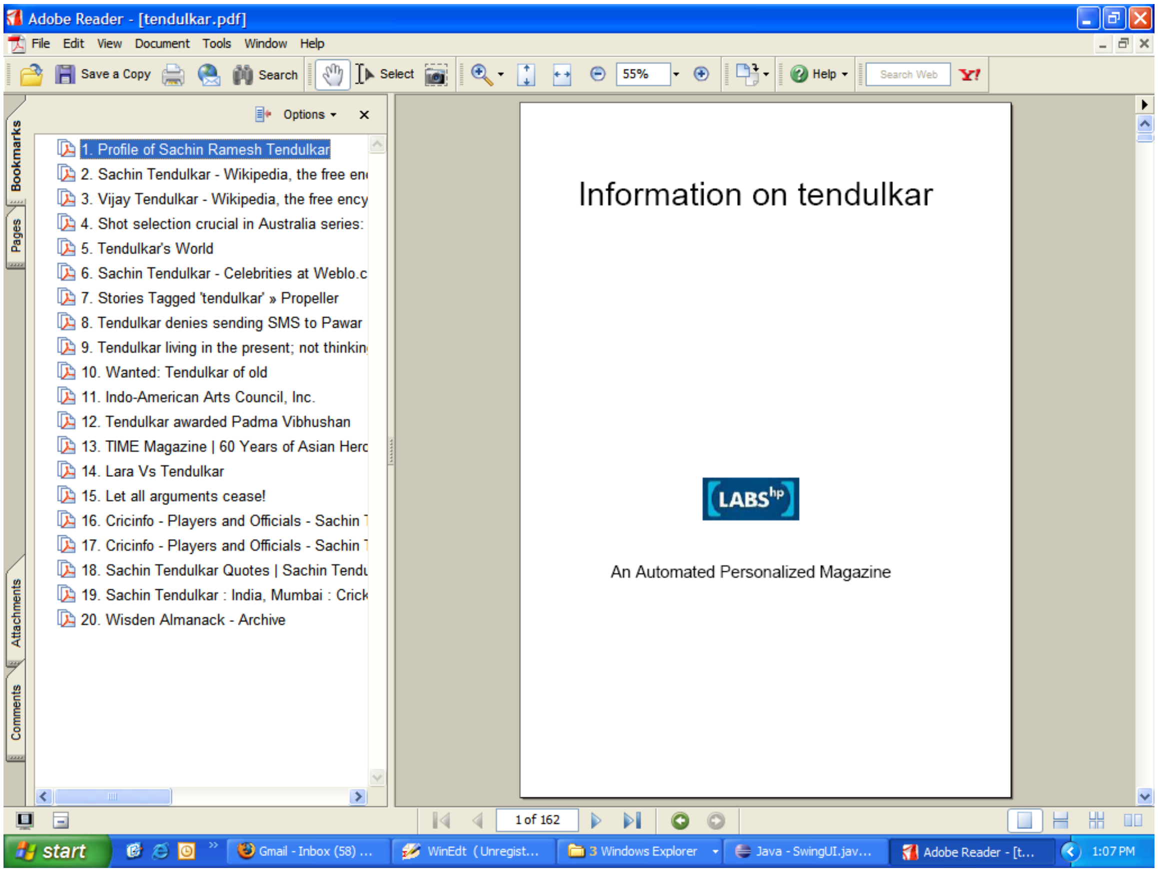


Figure 2: System Flowchart

The entire system was implemented in Java. The system is Platform independent and highly portable. It is currently implemented as a *desktop based live , 'on-the-fly search' search and pdf production mechanism* and can be easily extended into a web service by incorporating it as a servlet. The User Interface has been built using Java Swing. The User interface inputs the search query , depth of crawl and the number of results required by the user and produces a consolidated PDF document with the results (figure 3).



1. Profile of Sachin Ramesh Tendulkar
2. Sachin Tendulkar - Wikipedia, the free en
3. Vijay Tendulkar - Wikipedia, the free ency
4. Shot selection crucial in Australia series:
5. Tendulkar's World
6. Sachin Tendulkar - Celebrities at Weblo.c
7. Stories Tagged 'tendulkar' » Propeller
8. Tendulkar denies sending SMS to Pawar
9. Tendulkar living in the present; not thinkin
10. Wanted: Tendulkar of old
11. Indo-American Arts Council, Inc.
12. Tendulkar awarded Padma Vibhushan
13. TIME Magazine | 60 Years of Asian Herc
14. Lara Vs Tendulkar
15. Let all arguments cease!
16. Cricinfo - Players and Officials - Sachin T
17. Cricinfo - Players and Officials - Sachin T
18. Sachin Tendulkar Quotes | Sachin Tendu
19. Sachin Tendulkar : India, Mumbai : Crick
20. Wisden Almanack - Archive

Information on tendulkar

An Automated Personalized Magazine

Figure 3: First Page of the PDF Output

5 Evaluation

To evaluate our system, we conducted a user study to draw a comparison between the kinds of results given by our system as against those given by the general purpose search engines, viz. Yahoo! *[http://search.yahoo.com]* and Google *[http://www.google.com]*. As mentioned earlier, this system does not aim to replace the actual general purpose web-search engines. The experiment was carried out only to quickly gauge how relevant the users perceived the results produced by our system against those generated by the state-of-the-art search engines. The users were asked to rank the resultant URLs on a scale from 0 to 10 without revealing the source or the ranking of the URLs. The graphs (figure 4-6) for some of the queries show that the quality of the links produced by our system is comparable to those by the general purpose search engines. A more detailed user study is needed to evaluate the results provided by the system for coverage of various aspects of the topic and uniqueness of the results.

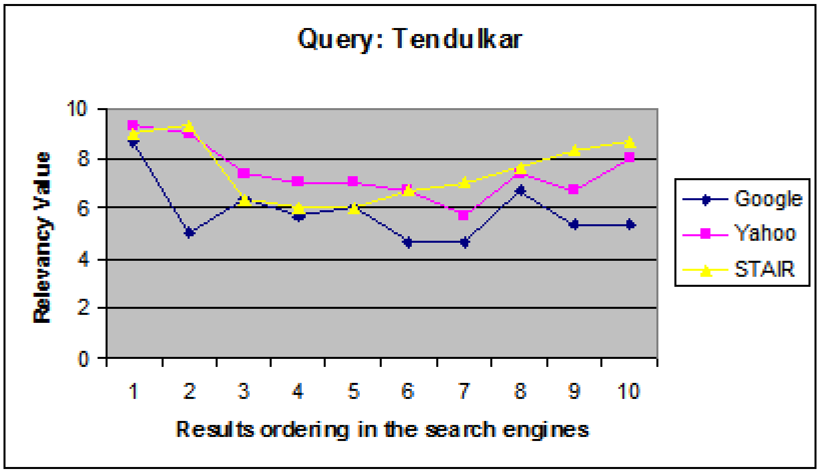


Figure 4: Results for the evaluation for the query: Tendulkar

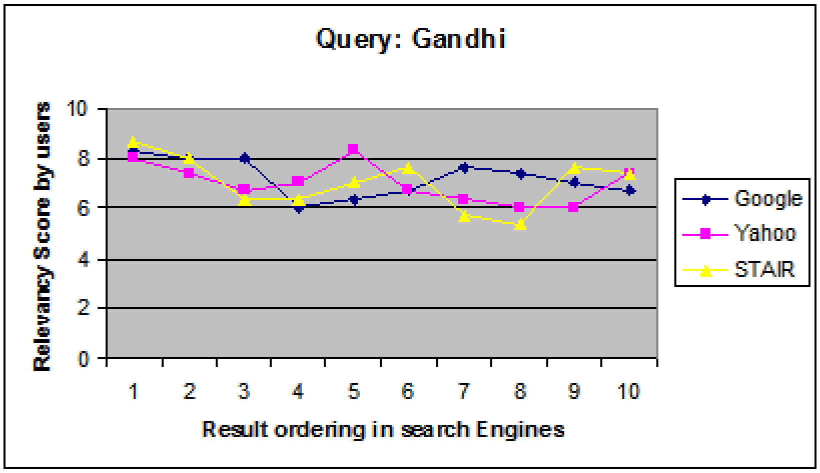


Figure 5: Results for the evaluation for the query: Gandhi

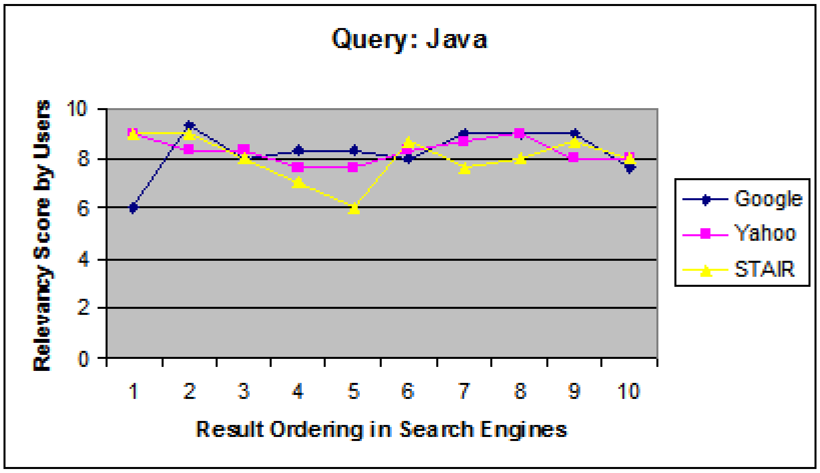


Figure 6: Results for the evaluation for the query: Java

6 Future work and Conclusions

The current system is a desktop based system that takes as its input an user query and gives as its output the consolidated pdf on a particular topic. We would like to improve the relevance of the results to the individual by using user profiles and context information. For example, a marketing executive searching for “mobile internet devices” is looking for very different information compared to a technical person. We plan to expand the user query by finding closely related topics.

Currently, we are not analyzing the content within the results, in future we would like to detect and eliminate very similar documents or documents that are a subset of other documents from the aggregated search results. Currently, Wordnet is used as the only ontology for extracting and placing synonyms of a word in the index. We plan to move one step ahead towards semantics based search by using the power of Wikipedia as the ontology in the Analysis stage. We also want to promote diversity in results, in some cases (e.g. reviews), we would like to detect opinions and provide results with differing opinions. Finally, we would like to detect facets of results and cluster and present the results in a faceted manner. On the user study side, a much bigger study is needed to assess the usefulness of the system.

In conclusion, in this work, we have present a prototype of a system for focused crawling, retrieval and presentation of consolidated user-specific information from the unstructured web. Our system, built over Apache Lucene , explores the web in a focused manner, guided by the relevance of the documents it finds. It filters the data at the data-acquisition level. Moreover, unlike the traditional search engines that provide just links, our system extracts, cleans and consolidates the content from the web into a PDF document, providing a novel user experience. The short user study indicated that the quality of the links provided by our system seems to be comparable to those of the general purpose search engines.

References

[1] Charu C. Aggarwal *Learning Strategies for Topic Specific Web Crawling* , Next generation data mining applications, 2004.

[2] Sergej Sizov, Michael Biwer, et al. *The BINGO! System for Information portal Generation and Expert Web Search*, CIDR Conference, 2003.

[3] Soumen Chakrabarti,Martin H. van den Berg, Byrom Dom . *Distributed Hypertext Resource Discovery through Examples*, VLDB Conference 1999.

[4] Michael Hersovici, Micahal Jacovi, et al. *Shark Search Algorithm. An application : Tailored Web Site Mapping*, Computer networks and ISDN

systems, Elsevier 1998.

[5] Sandeep Pandey and Christopher Olston *Crawl Ordering by Search Impact*, WSDM '08.

[6] Prabhakar Raghavan et al. *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[7] Shashank Pandit, Chris Olston *Navigation aided retrieval*, WWW conference, 2007, pp. 391-400.

[8] Ryen White et.al. *Enhancing web search by promoting multiple search engine use*, SIGIR 2008, pp.43-50.

[9] Oliver A. McBryan. *Tools for Taming the Web. First International Conference on the World Wide Web,* GENVL and WWWW CERN, Geneva (Switzerland), May 25-27 1994.

[10] S.Brin and L.Page *The anatomy of a large-scale hyper textual web search engine,* WWW 7, pages 107 - 117, 1998.

[11] Mehmet S.Aktas, Mehmet A. Nacar and Filippo Menczer *Using Hyperlink Features to Personalize Web Search*, LNCS 2006, Springer.