Identifying, Retrieving and Determining Relevance of Heterogenous Internet Resources

Emil Gatial and Zoltan Balogh

Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
emil.gatial@savba.sk

Abstract. This paper describes a chain of tools used for identifying, retrieving and determining relevance of heterogeneous Internet documents. RIDAR is a tool which acquires a set of relevant resource addresses capitalizing the potential of existing search engines such as Google or Yahoo through their web service interfaces. Identified Internet data resources are inserted into a shared database, which is then use by a tool called WebCrawler to retrieve relevant targets. WebCrawler repetitively retrieves the documents found at URL addresses identified by RIDAR. The crawling process employs the ERID (Estimate Relevance of Internet Documents) tool to gain relevance estimation used for downloading and crawling decisions. Such crawling process is called focused crawling. The tool downloads only documents described by the most specific set of keywords in the domain. The tools described in this article were designed and developed within the NAZOU project.

1 Tools chain

Following description of tools RIDAR, ERID and WebCrawler presents the chain of identifying, retrieving and determining relevance of heterogeneous Internet documents. The RIDAR tool exploits the potential of existing search engines to identify relevant information resouces on the Internet based on supplied serch terms or more complicated search expressions. Identified Internet data resources are inserted into a shared database, which is then use by a tool called WebCrawler to retrieve relevant targets. WebCrawler retrieves and processes documents from the Internet. Each of downloaded documents is analyzed using the ERID tool, which searches for specific keyword and tries to estimate the page relevance. According to such estimation WebCrawler decides whether the page is stored or not. Moreover, the result of ERID influences the search strategy during the crawling.

2 RIDAR

Information aquirying systems often require to identify primary internet resources. RIDAR allows to exploit existing search engines to retrieve links to

relevant Internet resources based on users-supplied serch terms or more complicated search expressions. Details about identified resources (URL, title, etc.) are stored into databases. The tool can integrate any search engine which exposes a web service API. Currently, RIDAR supports and had integrated the Google and Yahoo search engines. RIDAR provides generic interfaces which allow integrating search engines as well as targets for storing search results (databases). RIDAR allows storing retrieved results into a chosen target such as database or generic file. Currently MySQL target is implemented in RIDAR.

3 WebCrawler

Web is directed graph, where the nodes represent pages and the directed links represent hypertext links used to refer on the other pages in the Internet. Crawler process usually starts with a page, where it searches for hypertext links and explores the tree made of the hypertext links. Common crawler is usually composed of a data acquisition unit and scheduler unit. Data acquisition provides methods for Internet connection and data download, while the scheduler selects the URLs that have to be processed. Moreover, crawler employs the auxiliary queues and storage. The schema of a common crawler could be depicted like on the figure 1. Usually crawlers must conform to several policies, which considerable influence

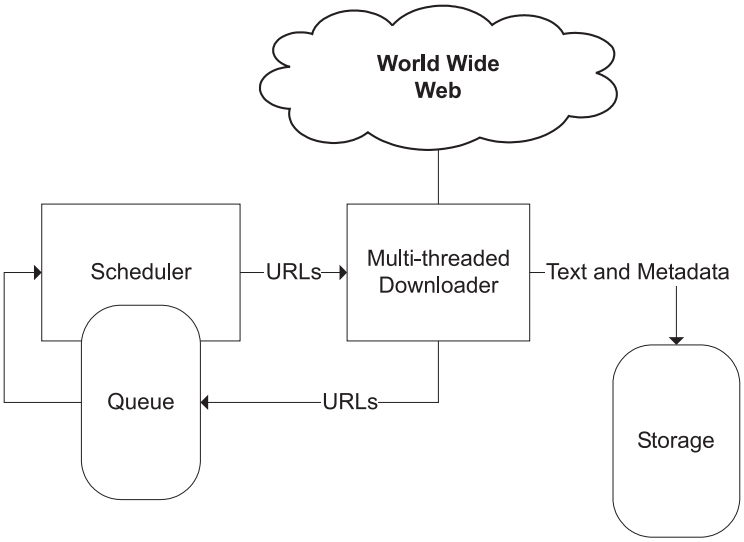

Fig. 1. Common crawler schema.

the crawler behavior.

3.1 Selection policy

Selection policies are based on graph exploration algorithms. The most common used search algorithms are breadth-first search (BFS), search with Page rank calculations, backlink count strategy, OPIC strategy [2]. Another possible search strategy is depth-firstsearch (DFS); although it is not widely used because of its high demand on data structureand processing. WebCrawler implements breadth-first strategy with the local site priority. This strategy continues selecting pages in breadth manner, but takes URLs from the queue of unprocessed URLs which points to the currently processing site. If the site is processed, it is add to the hash of processed sites. The links of the processed site are deleted and the processing continues with next unprocessed link. The next and the most important feature of WebCrawler is focused [1] crawling strategy for selecting of pages that will be processed. The ERID tool is used for importance estimation of downloaded pages, but its use is restricted up to specific level of crawling because the relevant pages aren't usually located in the top levels. The tests indicated that it is recommended to restrict relevance estimation at most in the fist two levels of processing site. WebCrawler restricts processing of non-HTML pages, although in the next implementation the non-HTML pages can be converted into plain-text using Converter tool and consequently processed by other tools.

3.2 Re-visit policy

If the WebCrawler processes a link which has already been processed it checks the age of local copy and refresh the page if the copy is outdated. Usually, the crawlers should periodically synchronize the web and the records about the stored documents. The record should be modified according to the current status of web, although in many cases it is difficult to identify the page modification, because of its dynamic content.

3.3 Politeness policy

Crawlers have to be aware of the network limits and servers overhead to restrict network overload and possibly avoid of crashing the processing site. The Web-Crawler can setup the access interval, but it is set to 0 seconds only for the testing purposes.

3.4 Parallelization policy

A parallel crawler is a crawler that runs multiple processes in parallel. The goal is tomaximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid of multiple downloading of the same page by different process, the assigning policy must be implemented. Cho and Garcia-Molina [3] examined two types of policies:

- Dynamic assignment: central server or process assigns the URLs for processingto the particular processes of the crawler dynamically. Central server can balanceload for every process, even though it may become the bottleneck of crawling.
- Static assignment: processing URLs are assigned before the process is created according to the hash function. This type of policy can unevenly share the load among the processes. On the other side the management of processes is very easy. WebCrawler doesn't implement the any parallelization policy in this stage, even though itis planned in the next version.

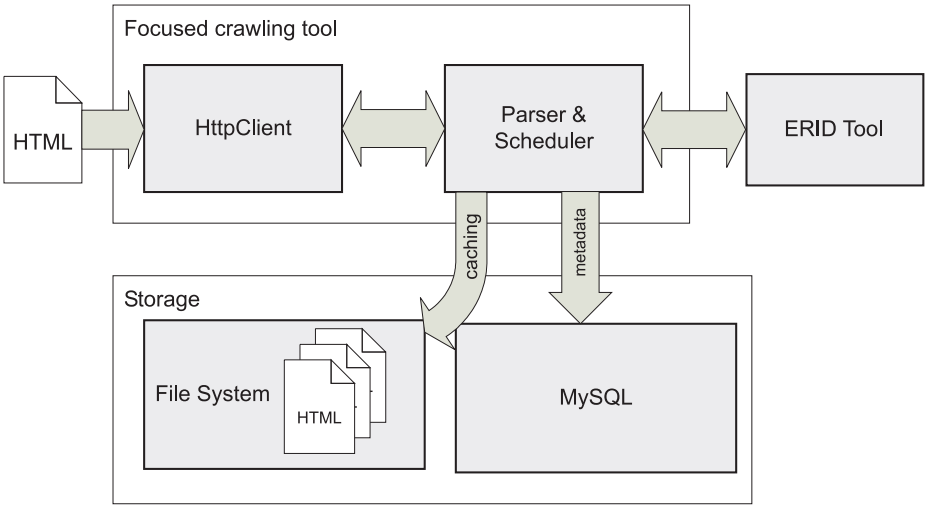

Fig. 2. Block scheme of WebCrawler tool.

The record about the download time, original URL and location of local page copy is created in database. WebCrawler tool updates these records anytime it passes through the already processed site. Other tools can access downloaded documents by using these records. The block schema is depicted in the figure 2.

3.5 Experimental results

Testing results were measured along with use of ERID tool. One process of the WebCrawler was used to process exactly one web site. In the test case, the connection time, transfer time and time of page processing were measured. The results are located in the table 1 and table 2. This testing reflects the network speed, connection and ratio of page processing and page download which is about 3%.

Table 1. Results of downloading and processing 350 pages for web site http://www.jobseeker.com/ using tools WebCrawler and ERID.

Connection time	124,728 [sec.]
Download time	84,613 [sec.]
Processing time	6,685 [sec.]
Overall time	219,633 [sec.]

Table 2. Results of downloading and processing 350 pages for web site http://www.theitjobboard.com/ using tools WebCrawler and ERID.

Connection time	654,310 [sec.]
Download time	27,133 [sec.]
Processing time	15,415 [sec.]
Overall time	725,088 [sec.]

4 ERID - Estimate Relevance of Internet Documents

The tool evaluates the content of arbitrary document and computes the relevance estimation. The relevance value is computed according to the term frequency found on the page. Term frequency [4] is defined as

$$t_i = \frac{n_i}{\sum_k n_k}, \tag{1}$$

where n_i being the number of occurrences of the considered term and the sum express the overall number of occurrences of every term. The keyword pool and its weights must by setup in advance. The relevance is counted asan average of term frequencies of given document and then document is evaluated relevant if it passes through the predefined threshold. The domain is configured using the most characteristic keywords. The domain of relevance is defined according to the keywords pattern specific for given domain. The relevance estimation tires to decrease amount of downloaded documents by eliminating the pages with uninteresting content. Because ofthe page content analysis is based on searching only for predefined keywords, the used method is quite fast and ensures that some keyword patterns will occur in downloaded page.

4.1 Experimental results

Testing results (see table 3 and table 4) were measured along with use of WebCrawler tool. One process of the WebCrawler was used to process exactly one web site. Results in columns express number of accepted and refused pages using ERID for relevance estimation. The relevance of each page was manually checked after the downloading of 350 pages. The row *Correct estimation* expresses the number of pages, which were successfully estimated by ERID, on contrary the

Table 3. Results of relevance estimation of 350 pages for processing for web site http://www.jobseeker.com/ by tools WebCrawler and ERID.

	Accepted	Refused
Correct estimation	287	48
Incorrect estimation	12	3
Overall number of downloaded pages	299	51

Table 4. Results of relevance estimation of 350 pages for processing for web site http://www.theitjobboard.com/ by tools WebCrawler and ERID.

	Accepted	Refused
Correct estimation	69	221
Incorrect estimation	60	0
Overall number of downloaded pages	129	221

row *Incorrect estimation* expresses the number of pages, which were unsuccessfully estimated by ERID. The goal of the ERID is to maximize the sum of correct estimated pages and minimize the number of pages which were refused and unsuccessfully estimated using ERID. The behavior of the tool is highly dependent on the keyword set and its weights.

5 Conclusion

The chain of tools were presented that models the relevancy estimation and crawling mechanism to restrict downloaded pages only from specific domain. The possible improvements of tools and work in the progess is sketched in the paper. Results of tools show that the estimation of page relevance and page processing is only small fraction regarding the page download time. Along with the effecient tool for relevance estimation this can considerably improve efeciency of crawling process. This chain of tools is used within the NAZOU project. Here, it provides raw data for information extraction using Ontea tool [5] which are subsequently presented in the Web [6].

Acknowledgements

This work was partially supported by the Slovak State Programme of Research and Development "Establishing of Information Society" under the contract No. 1025/04, K-Wf Grid EU RTD IST FP6-511385, and RAPORT APVT-51-024604 projects.

References

1. Diligenti, M., Coetzee, F., Lawrence, S., Giles, C. L., and Gori, M.: Focused crawling using context graphs. In: Proceedings of 26th International Conference on Very Large Databases (VLDB), pages 527-534, Cairo, Egypt, 2000.
2. Abiteboul, S., Preda, M., and Cobena, G.: Adaptive on-line page importance computation. In: Proceedings of the twelfth international conference on World Wide Web: pages 280-290, 2003
3. Cho, J., Garcia-Molina, H.: Parallel crawlers. In: Proceedings of the eleventh international conference on World Wide Web, pages 124-135, Honolulu, Hawaii, USA. ACM Press, 2003
4. Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5): pages 513-523, 1988
5. Laclavik, M., Seleng, M., Gatial, E., Balogh, Z., Hluchy, L.: Ontology based Text Annotation OnTeA; In: Proc. of 16-th European-Japanese Conf. on Information Mo delling and Knowledge Bases, EJC'2006, Y.Kiyoki et. al. eds., 2006, Dept.of Computer Science, VSB - Technical University of Ostrava, pp. 280-284, ISBN 80-248-1023-9. Trojanovice, Czech Republic.
6. Navrat, P., Bielikova M., Rozinajova, V.: Methods and Tools for Acquiring and Presenting Information and Knowledge in the Web. In: CompSysTech 2005, B. Rach ev, A. Smrikarov (Eds.), Varna, Bulgaria, June 2005. pp. IIIB.7.1–IIIB.7.6.