# SpidersRUs: Automated Development of Vertical Search Engines in Different Domains and Languages

Michael Chau
School of Business
The University of Hong Kong
Pokfulam, Hong Kong
+852 2859-1014
mchau@business.hku.hk

Jialun Qin, Yilu Zhou, Chunju Tseng, Hsinchun Chen
Department of Management Information Systems
The University of Arizona
Tucson, Arizona 85721, USA
+1 520-621-2748
{qin, yilu, chunju}@u.arizona.edu, hchen@eller.arizona.edu

## ABSTRACT
In this paper we discuss the architecture of a tool designed to help users develop vertical search engines in different domains and different languages. The design of the tool is presented and an evaluation study was conducted, showing that the system is easier to use than other existing tools.

## Categories and Subject Descriptors
H.3.7 [**Information Storage and Retrieval**]: Digital Libraries – *collection, system issues, user issues.*

## General Terms
Design, Experimentation, Human Factors.

## Keywords
Search engine, Web spiders, Web crawlers.

## 1. INTRODUCTION
Recently, many domain-specific or language-specific search engines have been built to facilitate more efficient searching in different areas. However, much effort will be needed to construct such a search engine that provides effective and efficient search functionalities with a high-quality collection of documents. Although comprehensive software tools for creating search engines exist, most of them cannot work on non-English languages such as Asian and Middle East languages. In this paper, we present our work in designing and implementing a system to address this problem.

## 2. RELATED WORK
Due to the large size of the Web, it is often desirable to build domain-specific or language-specific collections that can be searched and managed more easily. When developing Web search engines, "spiders" program are often used to traverse the Web to collect Web pages [2]. The spiders read from a list of starting seed URLs and download the documents at these URLs. The spiders then follow the links in these documents to continue the process until a required number of documents have been collected. Spidering tools have been available since the early days of the Web, e.g., tueMosaic [4]. Documents retrieved by spiders are stored in a repository of Web pages. An indexer processes the documents in the repository and builds an underlying index of the search engine. SMART and Glimpse [5] are examples of indexing tools that have been widely used. After an index is created, a query engine is responsible for retrieving and ranking documents from the index according to user queries. The results are usually rendered in an HTML format and sent back to the user through the Web.

There are many free software tools that provide all of the components necessary for building a search engine, i.e., collection building, indexing, searching, index storage structure, and user interface. Users can build their own search engines with such tools. Some popular examples of comprehensive search engine development tools are WebGlimpse [5], Greenstone [6], ht://dig, and Alkaline. However, most of these tools only work for English documents and are not able to process non-English documents, especially for non-alphabetical languages. Only a few of them, such as Greenstone, support multilingual collection building. As a result, most of these tools cannot be used to create digital library for non-English collections.

## 3. SPIDERSRUS: SYSTEM DESIGN
SpidersRUs (http://ai.bpa.arizona.edu/spidersrus/) is designed based on the backend architecture of popular Web search engines, but in a much simpler way [1]. The design is shown in Figure 1. The Spider module is responsible for collecting documents from the Web. This module contains multiple threads that work in parallel to download Web documents in a breadth-first search order. Users also can specify filters and term lists to guide the spiders. The downloaded documents are stored as files in their original form in the local repository. These files are then indexed by the Indexer module, which is designed to handle documents in different languages, such as English, Chinese, Arabic, Spanish, Japanese, and so on. The Indexer can also handle documents in formats other than HTML, such as PDF, Word, etc. Terms are extracted from the documents and the indexes are created and saved as files. After creating the indexes, the users can start the Searcher module which will load the files, allowing users to search the collection through the Web. For the detailed design of each module, users are referred to [3].

The SpidersRUs toolkit allows users to build, access, and maintain multiple document collections in multiple languages. The first step of building a new document collection is to create a new project in the toolkit. After a project is created, the user can interact with the Spider module by specifying the "seed URLs" and other parameters for the crawling process. After the spidering


*JCDL '05*, June 7–11, 2005, Denver, Colorado, USA

process is completed, the user can start indexing the documents collected. An inverted index will be created for the collection. The user can then assign a port number for the search service for the current collection. The user interface of the search engine and the result display format also can be customized by the user by modifying the HTML template and the image files used. After loading the service, the search interface to the collection will be available to any users connecting to the specified port from the Web. In Figure 2, we show the example of a search engine built by the toolkit in Chinese. As discussed earlier, the support for multiple languages is not available in most current search engine tools and will be very useful for vertical search engine development. The Web user interface is designed in English but it can be easily customized for different languages.

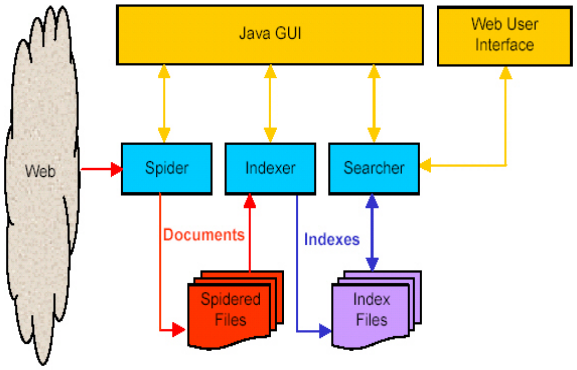


**Figure 1. The high-level architecture of SpidersRUs.**

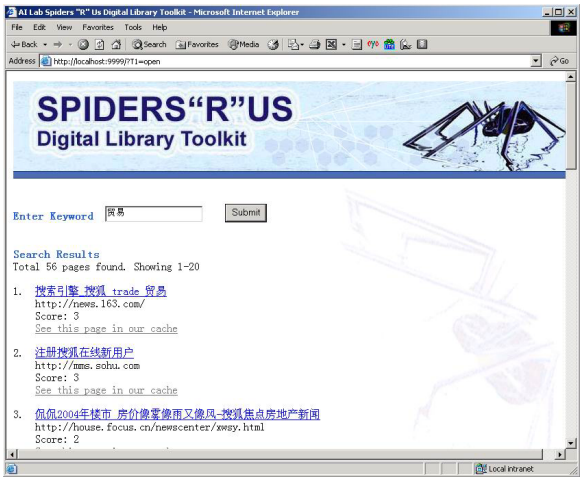


**Figure 2. Example of a Chinese search engine built by SpidersRUs.**

# 4. SYSTEM EVALUATION

To evaluate the tool, we conducted a study to compare it with two existing search engine development tools, namely Alkaline and Greenstone discussed earlier. A group of 28 business students were asked to use the three systems to build a vertical search engine in a domain they chose. After building the systems they were asked to fill in a questionnaire to evaluate the systems by answering a series of questions in the scale of 0 (lowest) to 9 (highest). Some of the data collected are shown in Table 1. When asked about the ease of use of the different tools in the process of building a specialized search engine, the subjects rated SpidersRUs an average score of 6.46, which is significantly better than Alkaline (5.32) and Greenstone (4.21) at the 5% level. In the second item "learning to operate the system", SpidersRUs also scored significantly better than Alkaline and Greenstone at the 1% level. For the other two measures, we observed that SpidersRUs scored significantly higher than Greenstone at the 1% level and comparable to Alkaline.

| | SpidersRUs | Alkaline | Greenstone |
|---|---|---|---|
| Overall ease of use of the system | 6.46 | 5.32 | 4.21 |
| Learning to operate the system | 6.75 | 5.14 | 4.79 |
| User's control over the process | 5.68 | 5.14 | 3.86 |
| Ease of specifying parameters | 6.29 | 6.04 | 4.21 |

**Table 1. Evaluation results.**

# 5. CONCLUSION

In this paper we discuss the architecture of a toolkit designed to help developers create vertical search engines in different domains and different languages. Our evaluation study showed that the tool was perceived better by users than some existing tools. We are currently expanding the tool for more functionalities, such as multimedia support.

# 6. ACKNOWLEDGMENTS

This project has been supported in part by the following grants: NSF Digital Library Initiative-2 IIS-9817473, NSF National SMETE Digital Library DUE-0121741, and HKU Seed Funding for Basic Research.

# 7. REFERENCES

[1] Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the Web. ACM Transactions on Internet Technology, 1(1) (2001) 2-43.

[2] Chau, M. and Chen, H.: Personalized and Focused Web Spiders. In: Zhong, N., Liu, J., Yao Y. (eds): Web Intelligence. Springer-Verlag (2003) 197-217.

[3] Chau, M., Qin, J., Zhou, Y., Tseng, C., and Chen, H.: A Framework for Creating Specialized Search Engines in Multiple Languages. Journal of the American Society for Information Science and Technology, forthcoming.

[4] DeBra, P. and Post, R.: Information Retrieval in the World-Wide Web: Making Client-based Searching Feasible. Proceedings of the First International World Wide Web Conference, Geneva, Switzerland (1994).

[5] Manber, U., Smith, M., and Gopal, B.: WebGlimpse: Combining Browsing and Searching. Proceedings of the USENIX Annual Technical Conference, California (1997).

[6] Witten, I. H., McNab, R. J., Boddie, S. J., and Bainbridge, D.: Greenstone: A Comprehensive Open-Source Digital Library Software System. Proceedings of the ACM Digital Libraries Conference, San Antonio, Texas, USA (2000).