MySearchView: A Customized Metasearch Engine Generator

Yiyao Lu[1,2], Zonghuan Wu[1,3], Hongkun Zhao[1,2], Weiyi Meng[1,2]
King-Lup Liu[1], Vijay Raghavan[1,3], Clement Yu[1,4]
[1]Webscalers, LLC, Lafayette, LA 70506, USA, kliu@webscalers.com
[2]SUNY Binghamton, Binghamton, NY 13902, USA, {ylu0,hkzhao,meng}@cs.binghamton.edu
[3]University of Louisiana at Lafayette, Lafayette, LA 70504, {zwu, vijay}@cacs.louisiana.edu
[4]University of Illinois at Chicago, Chicago, IL 60607, yu@cs.uic.edu

ABSTRACT
In this paper, we describe MySearchView – a system for assembling search engines into metasearch engines. With this system, any user can create a metasearch engine by simply letting the system know the URLs of the search engines the user wants to be included and the metasearch engine will be built fully automatically. In this paper, the main steps of building metasearch engines will be sketched. We will also outline our plan to demonstrate all the features of MySearchView.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *retrieval model*, *search process*.

General Terms
Management, Measurement, Performance, Design.

Keywords
Metasearch engine, customization, wrapper generation, result merging.

1. INTRODUCTION
A metasearch engine is a system that supports unified access to multiple existing search engines [1]. Metasearch engines have several advantages over regular search engines such as its ability to combine the search coverage of multiple search engines, its ability to reach the Deep Web, and its ability to obtain more relevant and more up-to-date results [1]. Building a high quality metasearch engine is a non-trivial process as it requires the developer to have deep understanding of several technologies.

2. CUSTOMIZED METASEARCH ENGINE
A customized metasearch engine is a metasearch engine that is tailored to the needs of a particular user or a particular group of users. Each metasearch engine is built on top of a set of existing search engines. In this paper, metasearch engine customization mostly means that we allow a user to choose the search engines for each of his/her metasearch engines.

We use two examples to illustrate how a customized metasearch engine may help satisfy the special search needs of a user.

SIGMOD'07, June 11–14, 2007, Beijing, China.
ACM 978-1-59593-686-8/07/0006.

Example 1. Jack is a picker of sports news for a newspaper's sports section. Jack has identified about twenty sports sites on the Web and he visits these sites every day to check if there are interesting sports stories to include for the newspaper. On each of these sports sites, one can find the news articles by submitting queries to the site search engine. If a metasearch engine can be built on top of the search engines of the twenty sports sites just for Jack, he would be able to spend significantly less time to gather his sports news everyday. Furthermore, the result-merging algorithm of the metasearch engine can help identify more significant news items by identifying those results that appear in the result lists of multiple local search engines. Research in information retrieval has shown that results that are returned by multiple approaches/systems are more likely to be relevant [12].

Example 2. Bill reads news on the Internet. Due to his strong political orientation, he only reads news and commentaries from certain newspapers that fit his tastes and politics. As a result, he does not use popular news search engines to find news articles to read because he hates to see articles from some undesirable sites and he does not want to go through a long list of results to find the articles from his desired sites. A dedicated metasearch engine that just connects to Bill's desirable news search engines can solve Bill's problem.

The above examples show that ordinary people from different walks of life can benefit from metasearch engines that are specifically built for their special search needs. In general, a customized metasearch engine can save users valuable time in search and can allow a user to search only trusted sources.

Most Web users do not have the expertise (e.g., programming skill, knowledge about HTML and HTTP) to build a metasearch engine by themselves and hiring experts to build these metasearch engines can be very expensive.

3. MYSEARCHVIEW
MySearchView is a tool that can automate the construction of metasearch engines. Any person who knows how to use search engines and knows what URLs are can create a metasearch engine using this tool. In this section, we describe how to use MySearchView to create and manage metasearch engines from a user's perspective.

3.1 Metasearch Engine Creation

From a pull-down menu on the GUI of MySearchView, the user selects the "Create New MSE" to start the metasearch engine creation screen (see Figure 1).

From this interface, the user can give a name to the metasearch engine he/she is about to create. The textbox labeled URL is for the user to enter the URL of a search engine. Suppose the URL of the search engine is www.yahoo.com. The user can either type in this URL or cut-and-paste it from the Yahoo homepage. By clicking the "ADD TO BASKET" button, this search engine is moved to "Your SEARCH ENGINE BASKET". The user can repeat this process to enter more search engines for this metasearch engine. Once all of the desired search engine URLs are entered, the user can select one of the following two optional ways to display the search results.

1. Merge: With this option, for each user query submitted to this metasearch engine, the results returned from different search engines will be merged and the merged results will be displayed to the user. This is the default option. See Section 4 for more information about the merging algorithm.
2. Non-Merge: With this option, the search results from different search engines will be displayed separately. In addition, the number of hits for the user query will be displayed for each search engine.

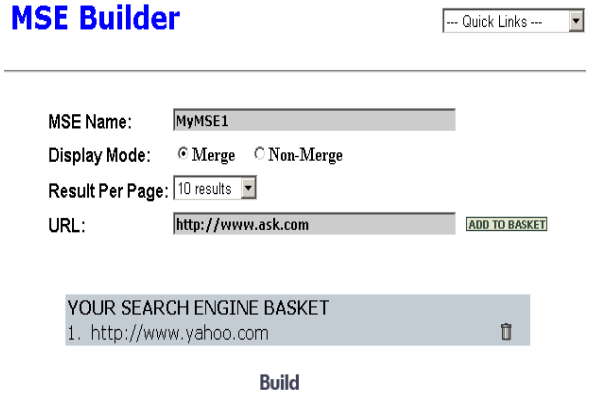

Figure 1. Metasearch Engine Creation Interface

Next, the user selects the number of results to be displayed on each result page of the metasearch engine from several options (10, 20, 50, 100; 10 is the default).

After all the options are selected, the user can click the "Build" button to start the creation of the metasearch engine.

Note that there is no guarantee that every search engine the user entered can be successfully incorporated into the metasearch engine. Among the reasons that may cause failure include: the entered URL is not valid, the Web page with the URL does not contain a search engine form, the search engine is down, and so on. For each search engine URL, MySearchView displays to the user whether or not the search engine is successfully incorporated. It takes an approximately one minute to incorporate each search engine.

A user may create multiple metasearch engines and specify one for them as the default (the one that is displayed after logging in).

3.2 Metasearch Engine Modification

MySearchView allows a user to modify any of his/her metasearch engines. Figure 2 shows the interface for metasearch engine modification. A user may change a metasearch engine in various ways: change its name, change the result display mode, change the number of results on a result page, remove one or more search engines, and add new search engines.

Figure 2. Metasearch Engine Update Interface

4. TECHNIQUES BEHIND THE SCENE

While it is fairly easy to use MySearchView to create metasearch engines automatically, the techniques that power it are quite complex. Due to limited space, we can only outline the technical challenges and provide a brief summary of the techniques here.

There are three main technical challenges:

1. *Automatic search engine connection.* This is to automatically create a program that can interact with the server of any given search engine, in both sending queries and receiving returned result pages. To achieve this, the Web page with the given URL needs to be parsed and the *search form* of the search engine needs to be analyzed to identify and extract all the information that enables automatic connection. In the simplest case, the name of the search engine server and its path can be identified from the "action" attribute of the form tag, the HTTP request method (either GET or POST) supported by the search engine can be obtained from the "method" attribute, and the query name that will hold the query string can be obtained from the "name" attribute. With these three pieces of information, a connection program can be generated automatically. Many difficulties may arise when dealing with more complex Web pages and some of them are: How to differentiate search forms from non-search forms? How to deal with multiple search forms on a single page? How to deal redirection? etc. Some of these issues were discussed in [3, 4].
2. *Automatic search result extraction.* When a result page is returned from a search engine, the search result records (SRRs) need to be extracted and other irrelevant information such as advertisements need to be discarded. Each SRR corresponds to a retrieved web page. A typical SRR consists of a title, a snippet, and a link to the web page. The challenge is to automatically generate the rules (wrapper) to extract the SRRs from the returned result pages of any given search engine. MySearchView employs the ViNTs system [5] to generate the wrapper automatically. ViNTs generates the wrapper for a search engine using several sample result pages that are

automatically collected from the search engine using automatically generated sample queries and the search engine connection mechanism identified above. Each sample result page will be rendered and its DOM tree will be generated so that its visual information (such as location of each node) and tag structure can be utilized to induce the record extraction wrapper for the page. The wrappers obtained based on different result pages from the same search engine may be somewhat different due to the possible variations of the displayed records. For example, Google displays some records un-indented while some indented. So it is possible that one sample result page has only un-indented SRRs while another sample result page has both indented and un-indented SRRs. The wrappers obtained based on result pages are then integrated to produce a robust wrapper. Please see [5] for more details about ViNTs.

3. *Result merging*. If a user submits a query to a metasearch engine that has the "Merge" display mode, the results (i.e., SRRs) returned from different search engines need to be merged. The quality of the merging algorithm employed by a metasearch engine probably has the most direct impact on the effectiveness of the metasearch engine. Many result-merging algorithms have been proposed (e.g., [6]). The merging algorithm employed by MySearchView is adopted (with revision) from a proprietary algorithm developed by Webscalers for its AllInOneNews news metasearch engine (www.allinonenews.com). Basically, this algorithm compares the user query with the title and snippet contained in each SRR and how well they match determines the rank of the SRR.

5. DEMO PLAN

We plan to demonstrate the entire process of creating and modifying metasearch engines. To allow users to truly appreciate the system, real search engines will be used. During the demo, we will try to engage the visitor(s) in an interactive mode. For example, we will invite the visitor(s) to provide the search engines and to make the choices on the result display mode and the number of results to display on each page. While the metasearch engine is being created (it takes a few minutes), we will explain the issues of automatic search engine connection and automatic wrapper generation. Once a metasearch engine is created, we will invite the visitor(s) to try it with their own queries. We will explain approximately how the result-merging algorithm works while viewing the search results. Then we will invite the visitor(s) to modify the metasearch engine and see how the modification takes effect.

6. RELATED WORK

We first note that MySearchView is not a metasearch engine but an automatic metasearch engine generation system. We are not aware of any similar tools.

Result merging has always been a critical component of metasearch engines and it has received extensive study (e.g., [1, 7, 8]). In the earlier days, merging algorithms were based on aggregating the local ranking scores and local ranks of the retrieved results. In recent years, as most search engines return the titles and snippets of the retrieved documents, more and more merging algorithms, including the one employed by MySearchView, utilize such information [6, 7, 8].

In the past, wrappers used to extract desired information from web sites were mostly generated manually or semi-automatically. In recent years, automatic wrapper generation techniques have received a lot of attention (e.g., [9, 10, 11]). ViNTs is the first wrapper generation system that utilizes both visual contents and tag information for wrapper generation and it is also one of the most accurate. ViNTs has been used to produce thousands of wrappers for many metasearch engines.

A previous version of the system was demonstrated at the ACM SIGIR conference in 2003 [2]. MySearchView is significantly different from the previous version. The search engine connection component has been largely improved. The wrapper generation component and result-merging component are completely new. Options such as the merging mode and the number of results to display on each result pages were not available in the previous version. Finally, the interface of MySearchView is completely redesigned to improve ease of use.

7. ACKNOWLEDGMENTS

This work is support in part by the following grants from NSF: SBIR grants (Phase I: DMI-0340348 and Phase II: DMI-0522271); and research grants: IIS-0208574, IIS-0208434, IIS-0414981, IIS-0414939, and CNS-0454298.

8. REFERENCES

[1] W. Meng, C. Yu, and K. Liu. Building Efficient and Effective Metasearch Engines. *ACM Computing Surveys*, 34(1), March 2002, pp.48-89.

[2] Z. Wu, V. Raghavan, et al. SE-LEGO: Creating Metasearch Engine on Demand. *ACM SIGIR*, Demo, 2003.

[3] Z. Wu, V. Raghavan, et al. Towards Automatic Incorporation of Search Engines into a Large-Scale Metasearch Engine. *IEEE/WIC WI Conference*, pp.658-661, 2003.

[4] J. Cope, N. Craswell, D. Hawking. Automated Discovery of Search Interfaces on the Web. ADC, 2003, pp.181-189.

[5] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu. Fully Automatic Wrapper Generation for Search Engines. *World Wide Web Conference (WWW14)*, pp.66-75, 2005.

[6] Y. Lu, W. Meng, L. Shu, C. Yu, and K. Liu. Evaluation of Result Merging Strategies for Metasearch Engines. *WISE Conference, 2005*.

[7] T Tsikrika, M Lalmas. Merging techniques for performing data fusion on the web. CIKM, 2001.

[8] Y Rasolofo, D Hawking, J Savoy. Result merging strategies for a current news metasearcher. *Information Processing and Management*, 2003.

[9] D. Buttler, L. Liu, C. Pu. A Fully Automated Object Extraction System for the World Wide Web. ICDCS, 2001.

[10] B. Liu, R. Grossman and Y. Zhai. Mining Data Records in Web Pages. SIGKDD, 2003.

[11] K. Simon, G. Lausen. ViPER: Augmenting Automatic Information Extraction with Visual Perceptions. CIKM, 2005.

[12] J. H. Lee. Analyses of Multiple Evidence Combination. ACM SIGIR, 1997.