Crawling the Hidden Web

(Extended Abstract)

Sriram Raghavan Hector Garcia-Molina
Computer Science Department
Stanford University
Stanford, CA 94305, USA
{rsram,hector}@cs.stanford.edu

ABSTRACT

Current-day crawlers retrieve content from the publicly indexable Web, i.e., the set of web pages reachable purely by following hypertext links, ignoring search forms and pages that require authorization or prior registration. In particular, they ignore the tremendous amount of high quality content "hidden" behind search forms, in large searchable electronic databases. Our work provides a framework for addressing the problem of extracting content from this hidden Web. At Stanford, we have built a task-specific hidden Web crawler called the Hidden Web Exposer (HiWE). In this poster, we describe the architecture of HiWE and outline some of the novel techniques that went into its design.

Keywords

Crawling, Hidden Web, Content extraction, HTML Forms

1. INTRODUCTION

A number of recent studies [1, 2, 3] have noted that a tremendous amount of content on the Web is *dynamic*. However, since current-day crawlers only crawl the *publicly indexable Web* [2], much of this dynamic content remains inaccessible for searching, indexing, and analysis. The hidden Web is particularly important, as organizations with large amounts of *high-quality* information (e.g., the Census Bureau, Patents and Trademarks Office, News media companies) are placing their content online, by building Web query front-ends to their databases.

Crawling the hidden Web is a very challenging problem for two fundamental reasons: (1) scale (a recent study [1] estimates the size of the hidden Web to be about 500 times the size of the publicly indexable Web) and (2) the need for crawlers to handle search interfaces designed primarily for humans.

We address these challenges by adopting a *task-specific human-assisted* approach to crawling. Specifically, we selectively crawl portions of the hidden Web, extracting content based on the requirements of a particular application or domain. We also provide a framework that allows the human expert to customize and assist the crawler in its activity.

2. HIWE

At Stanford, we have built a task-specific hidden Web crawler called the Hidden Web Exposer (HiWE). Figure 1 illustrates HiWE's architecture and execution flow.

Since search forms are the entry-points into the hidden Web, HiWE is designed to automatically process, analyze,

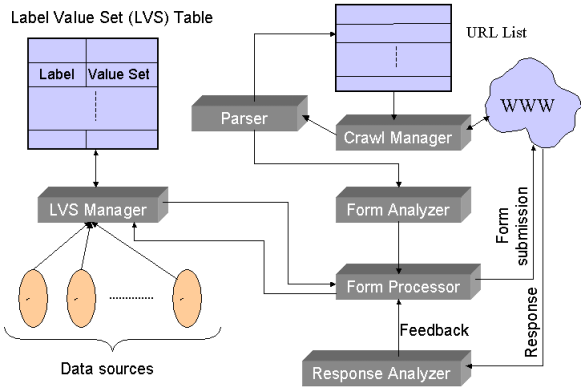

Figure 1: HiWE Architecture

and submit forms, using an internal model of forms and form submissions. This model treats forms as a set of *(element, domain)* pairs. A form element can be any one of the standard input objects such as selection lists, text boxes or radio buttons. Each form element is associated with a finite or infinite domain and a text *label* that semantically describes the element (see Figure 2).

The values used to fill out forms are maintained in a special table called the LVS (Label Value Set) table (Figure 1). Each entry in the LVS table consists of a label and an associated *fuzzy/graded set* of values (e.g., Label = "State" and value set = { ("California", 0.8), ("New York", 0.7) }). The weight associated with a value represents the crawler's estimate of how effective it would be, to assign that value to a form element with the corresponding label. Methods to populate the LVS table and assign weights are described in detail in [4].

The basic actions of HiWE (fetching pages, parsing and extracting URLs, and adding the URLs to a URL list) are similar to those of traditional crawlers. However, whereas the latter ignore forms, HiWE performs the following sequence of actions for each form on a page:

1. *Form Analysis:* Parse and process the form to build an internal representation based on the above model.

2. *Value assignment and ranking:* Use approximate string matching between the form labels and the labels in the LVS table to generate a set of candidate value as-

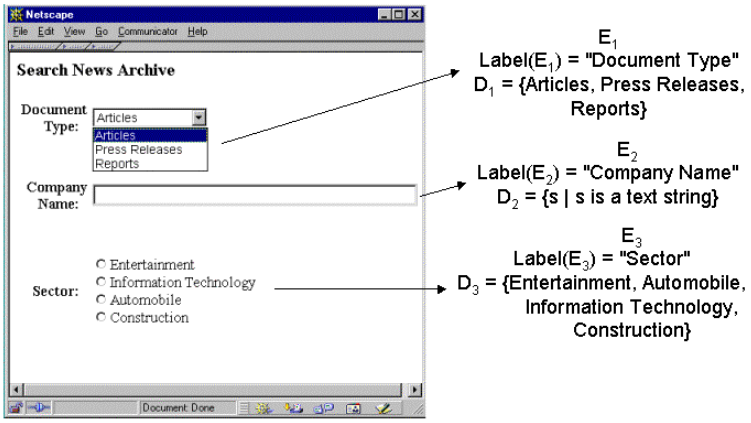

Figure 2: Sample labeled form

signments.[1] Use fuzzy aggregation functions to combine individual weights into weights for value assignments and use these weights for ranking the candidate assignments.

3. *Form Submission:* Use the top "N" value assignments to repeatedly fill out and submit the form.

4. *Response Analysis and Navigation:* Analyze the response pages (i.e., the pages received in response to form submissions) to check if the submission yielded valid search results. Use this feedback to tune the value assignments in Step 2. Crawl the hypertext links in the response page to some pre-specified depth.

3. LAYOUT-BASED EXTRACTION

Search forms and response pages are designed for human consumption. As a result, it is a significant task for a crawler to process and extract semantically useful information (e.g, the labels of form elements) from such pages.

As part of form and response analysis, HiWE uses a "Layout-based Information Extraction Technique (LITE)" to achieve this task. LITE is based on the principle that semantic information can be robustly extracted by exploiting visual cues (i.e., using information about how various objects are laid out on a page). For example, the label associated with a given form element is most likely to be the piece of text or phrase that is visually (not necessarily textually) closest to the form widget, when the page is displayed by the browser. Hence, HiWE employs a custom layout engine that approximately lays out form pages and response pages and can be used to compute visual distances between different elements in a page. Our preliminary experiments indicate a 93% success rate in using LITE to correctly identify labels for form elements.

4. CONCLUSION

We have addressed the problem of crawling and extracting content from the "hidden Web", the portion of the Web hidden behind searchable HTML forms. Due to the tremendous size of the hidden Web, comprehensive coverage is very difficult, and possibly less useful, than task-specific crawling. Our work exploits this specificity to design a configurable crawler that can benefit from knowledge of the particular application domain.

Our initial experiments [4] indicate that human-assisted crawling of the hidden Web is feasible. They also indicate that LITE is a powerful method for extracting semantic information from search forms and response pages.

[1] A *value assignment* is an assignment of a value to each element of a form.

5. REFERENCES

[1] The Deep Web: Surfacing Hidden Value. http://www.completeplanet.com/Tutorials/DeepWeb/.

[2] S. Lawrence and C. L. Giles. Searching the World Wide Web. *Science*, 280(5360):98, 1998.

[3] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Nature*, 400:107–109, 1999.

[4] S. Raghavan and H. Garcia-Molina. Crawling the hidden web. Technical Report 2000-36, Computer Science Department, Stanford University, December 2000. Available at http://dbpubs.stanford.edu/pub/2000-36.