Testbed for Information Extraction from Deep Web

Yasuhiro Yamada
Department of Informatics, Kyushu University
Fukuoka 812-8581, Japan
yshiro@matu.cc.kyushu-u.ac.jp

Nick Craswell
CSIRO Mathematical and Information Sciences
GPO Box 664, Canberra, ACT 2601, Australia
Nick.Craswell@cmis.csiro.au

Tetsuya Nakatoh
Computing and Communications Center,
Kyushu University
Fukuoka 812-8581, Japan
nakatoh@cc.kyushu-u.ac.jp

Sachio Hirokawa
Computing and Communications Center,
Kyushu University
Fukuoka 812-8581, Japan
hirokawa@cc.kyushu-u.ac.jp

ABSTRACT

Search results generated by searchable databases are served dynamically and far larger than the static documents on the Web. These results pages have been referred to as the Deep Web [1]. We need to extract the target data in results pages to integrate them on different searchable databases. We propose a testbed for information extraction from search results. We chose 100 databases randomly from 114,540 pages with search forms. Therefore, these databases have a good variety. We selected 51 databases which include URLs in a results page and manually identify target information to be extracted. We also suggest evaluation measures for comparing extraction methods and methods for extending the target data.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: Performance evaluation (efficiency and effectiveness)

General Terms

Experimentation

Keywords

Deep Web, Meta Search, Testbed, Wrapper

1. INTRODUCTION

An increasing number of specialized searchable databases are available online. Often significant effort has been put into creating such databases, which might contain listings of government information, jobs, movies, books, art holdings or products.

One way of accessing databases is through a search wrapper, which allows querying and retrieval of database contents. The information extracted from search results can be presented integrated with other results in a metasearcher, for example [3, 5], or added to a knowledge base or integrated database. The latter process of data gathering is known as Web mining.

Usually search results contain repetition of results in a particular format. Therefore a basic problem of metasearch or query-based Web mining is result extraction: to identify the format of results for a particular database and extract key data from each result such as its URL. Automating result extraction makes systems easier to create and more robust to changes in database presentation formats. A program to do this process is called a wrapper. Recently a lot of wrapper generation programs have been proposed [4].

In this poster we present a testbed for result extraction wrappers. It includes frozen results pages from 51 engines selected randomly from 114,540 pages with search forms and manually identified target information to be extracted. We also suggest evaluation measures for comparing wrapper methods and methods for extending the target data.

2. THE TESTBED

The testbed includes the query page to input keywords and results page of 51 databases, plus the manually verified target information, which should be extracted from each results page. The target information always includes the URL of each result (in this testbed we exclude databases which do not return result URLs).

2.1 Choosing the databases

The first key aspect in creating such a testbed is to be confident that a good variety of databases are represented. The testbed should be large and representative enough to include examples of important varieties of database, including keyword search engines, shopping sites, metasearch results and database searches which return tables of matching data.

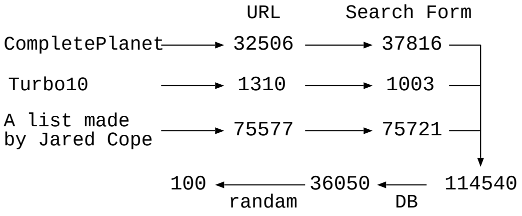

Figure 1: Choosing the databases

To get this variety we identified 114,540 search forms using list-

WWW2004, May 17–22, 2004, New York, NY USA.
ACM xxx.xxx.

ings from "Turbo10 Search Engine [3][1]," "CompletePlanet[2]" and a listing made by Jared Cope [2]. These forms targeted 36,050 separate database URLs, from which we randomly chose 100 (see Figure 1). Choosing 100 databases is better than choosing 10 or 20, because in that case we might miss out on some important major variety of database.

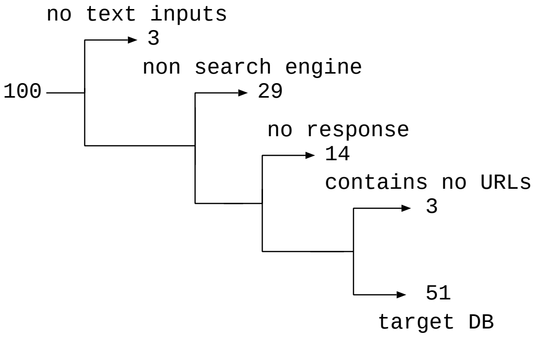

Figure 2: Eliminating selected databases

From our randomly selected 100, we eliminated 49 databases for various reasons (see Figure 2). These were databases without a text input field, which did not seem to be a search engine, which did not respond to queries and those whose results page contained no URLs.

Through manual querying we generated 5 results pages for each of the 51 remaining databases. There are 4 non English databases in all 51 databases. Each results page, which averaged 29 kilobytes in size, was saved along with its URL. Each results page contained on average 18 URLs.

2.2 Identifying target data

The other key aspect is to identify the target of extraction for database search results pages. Because of the variety of databases, it is difficult to extract all fields of every result format. For example, a book search might have author, title, summary and price fields while an email archive search has from, to and subject. It is possible to extract all fields, but in this study we instead concentrate on three main aspects:

1. Extracting the URLs from each search result. In finding our set of 100 databases, we only found 3 which did not have result URLs which link to more detail on the search result. Extracting the URLs of all results is therefore a general and basic problem.
2. Identifying the extent of the first search result. For the result at rank one we extract from the first word or tag to the last word or tag. Identifying the extent of search results is also a general and basic problem.
3. Extracting the number of records. All selected databases return a list of records in a result page. Therefore extracting the number of records is also a general and basic problem.

2.3 Evaluation measures

An important part of the testbed is evaluation measures. We can make some evaluation measures by using the testbed.

1. URL Extraction: The question is how many a wrapper can correctly extract URLs linking to more detail on the search result. We measure it by using the 1st identified information in Sec. 2.2. Let N be the number of URLs identified in testbed, R be the number of result URLs listed by a wrapper extraction, and n be the number of URLs correctly identified by a wrapper extraction. The precision of extraction is n/N and the recall is n/R.
2. Record Extraction: The question is how many a wrapper can correctly extract records. However it is difficult to measure all records in all pages. We measure it by using the 2nd and 3rd identified information in Sec. 2.2. It is more likely that a wrapper can extract all records if it can extract the first record and the number of records in each file.

For identifying the record extraction, we measure "success rate" out of 51 databases. Let x be the number of result pages in testbed, Let y be the number of result pages which a wrapper can correctly identified the first record and the number of records. The success rate is y/x.

3. EXTENDING THE TARGET DATA

The precise target of extraction for a wrapper depends on the application in question. We have initially chosen the result URL list, the extent of the first result and the number of records as the most important target data. However, it is possible to extend the testbed by identifying further target data. For example, some wrapper systems extract the title and summary from each result. To evaluate the effectiveness in extracting results counts, it is necessary to manually identify further target information.

We have done this, finding such information in most pages. This target data is available along with the rest of the data. We divide the first result in each result page into some fields, for instance ranking, title, summary, page size and so on. And we name each field a proper word. If the name of a field is written in the page, we use it. If not, we name it a proper word manually.

We can make a evaluation measure to extract fields as follows. The question is how many a wrapper can correctly extract fields. However fields to be extracted depend on systems. First, testbed users decide fields to be extracted. Let x be the number of fields to be extracted, and y be the number of fields which a wrapper can correctly identified the fields. The success rate is y/x.

4. CONCLUSION

We have identified a test set of databases representative of the population of searchable Web databases. For these, we saved results page and identified target data. This testbed is available at `http://daisen.cc.kyushu-u.ac.jp/TBDW/`.

5. REFERENCES

[1] BrightPlanet, The Deep Web: Surfacing Hidden Value, BrightPlanet White Paper, 2000.

[2] J. Cope, N. Craswell and D. Hawking, Automated Discovery of Search Interfaces on the Web, Proc. of *the 14th Australasian Database Conference*, pp. 181–189, 2003.

[3] N. Hamilton, The Mechanics of a Deep Net Metasearch Engine, Proc. of *the 12th International World Wide Web Conference*, 2003.

[4] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva and J. S. Teixeira, A Brief Survey of Web Data Extraction Tools, SIGMOD Record, Vol. 31, No. 2, pp. 84–93, 2002.

[5] T. Taguchi, Y. Koga and S. Hirokawa, Integration of Search Sites of the World Wide Web, Proc. of *the International Forum cum Conference on Information Technology and Communication*, Vol. 2, pp. 25–32, 2000.

[1] `http://turbo10.com/`

[2] `http://www.completeplanet.com/`