Automatically Constructing Collections of Online Databases

Extended Abstract

Luciano Barbosa
School of Computing, University of Utah
lbarbosa@cs.utah.edu

Juliana Freire
School of Computing, University of Utah
juliana@cs.utah.edu

Categories and Subject Descriptors: H.4 [Information Systems Applications]: Information Search and Retrieval

General Terms: Algorithms, Design.

Keywords: Online databases, modular classifiers, focused web crawling.

1. INTRODUCTION

Due the the explosion in the number of online databases, there has been increased interest in leveraging the high-quality information present in these databases [6, 1, 8]. However, finding the right databases can be very challenging. For example, if a biologist needs to locate databases related to molecular biology and searches on Google for the keywords "molecular biology database" over 27 million documents are returned. Among these, she will find pages that contain databases, but the results also include a very large number of pages from journals, scientific articles, etc.

Recognizing the need for better mechanisms to locate online databases, people have started to create online database collections such as the Molecular Biology Database Collection [4], which lists databases of value to biologists. This collection, however, has been manually created and is manually maintained by the National Library of Medicine. Since it is estimated that there are over 20 million online databases [6], manual approaches to this problem are not practical. Besides, since new databases are constantly being added, the coverage of a manually maintained collection can be greatly compromised.

In this paper, we describe a new approach to the problem of automatically locating and organizing online databases that belong to a given domain. There are a number of issues that make this problem particularly challenging. Since online databases are sparsely distributed on the Web, an efficient strategy is needed to locate the forms that serve as entry points to these databases. In addition, online databases do not publish their schemas and their contents are often hard to retrieve. Thus, a scalable solution must determine the relevance of a form to a given database domain by examining just the information available in (and around) the form. As shown in Figure 1, our framework for constructing topic-specific online database collections consists of two key components that address these problems: a focused crawler that efficiently locates pages that contain forms; and a form-filtering process that identifies forms that belong to a given database domain. Below, we give a brief overview of these components. For a more detailed description, see [2].

CIKM'06, November 5–11, 2006, Arlington, Virginia, USA.
ACM 1-59593-433-2/06/0011.

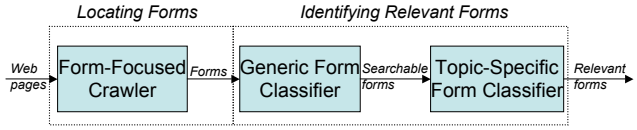

Figure 1: Organizing Online Databases.

2. LOCATING ONLINE DATABASES

A naïve approach to the problem of locating forms would be to visit all Web pages and extract the forms in these pages. Because the Web is large and forms are sparsely distributed, this solution is highly-inefficient—too many pages are visited unnecessarily. Besides, an exhaustive crawl can take weeks and this limits the ability to maintain the form set up-to-date.

A more efficient alternative is to use a focused crawler. In our solution, we use the Form Focused Crawler (FFC) [1]. The FFC is trained to efficiently locate forms that serve as the entry points to online databases—it focuses its search by taking into account both the *contents of pages* and *patterns in and around the hyperlinks in paths to a Web page*. Similar to focused crawlers (see e.g., [3]), the FFC focuses the crawl on a given topic—it uses classifier which, based on the contents of pages, guides the crawler to focus the search on pages that belong to a specific topic. To further focus the crawl, the FFC judiciously prioritizes links to follow that are more likely to lead to pages that contain forms—it does so by learning patterns of links that lead to pages which contain forms in a given database domain. An experimental evaluation showed that the FFC is more efficient (up to an order of magnitude) than a set of representative crawlers. For more details about the FFC, the reader is referred to [1].

3. IDENTIFYING RELEVANT FORMS

Although the FFC is trained to focus its crawl on a particular topic and database domain, the set of forms it retrieves is highly heterogeneous. Some forms are non-searchable, i.e., they do not correspond to database queries. Examples of non-searchable forms include forms for login, mailing list subscriptions, quote requests. Other forms, although searchable, may belong to different databases domains. The

(a) Job search forms

(b) Hotel and Aifare search forms

Figure 2: Variability in Web Forms. (a) Forms in Job domain with different attribute names representing the same concepts; (b) forms in two distinct domains, Hotel and Airfare, which contain attributes with similar labels.

problem is further complicated by the fact that there can be high variability in the contents and structures of forms that belong to a domain, as well as high similarity between forms in different domains.

The form-filtering component of our framework aims to select from the set retrieved by the crawler only *relevant* forms. More precisely, the problem we are trying to solve can be stated as follows: *Given a set F of heterogeneous, automatically gathered Web forms and an online database domain D, our goal is to select from F only the forms that are entry points to databases in D.* In other words, we would like to filter out all *irrelevant forms*—the non-searchable forms and searchable forms that do not belong to the domain D.

Since our goal is to devise a general solution to this problem, that works across different domains, we formalize the problem of identifying relevant forms in a particular database domain in terms of inductive learning concepts [7]. But unlike previous works on form classification which built monolithic classifiers, our approach is based on a divide-and-conquer strategy. Instead of using a single, complex classifier, our form filtering process uses two simpler classifiers that learn patterns of different subsets of the form feature space.

The *Generic Form Classifier* (*GFC*) learns patterns of structural features of forms, such as, for example: number of hidden tags; number of radio tags; number of file inputs; number of submit tags; number of image inputs; number of buttons; number of resets; number of password tags; number of textboxes; number of items in selection lists; sum of text sizes in textboxes; submission method (post or get). Empirically, we have observed that these structural characteristics of a form are a good indicator as to whether the form is searchable or not [1].

The GFC is effective for identifying searchable forms, regardless of their domains. However, even when a focused crawler is used, the set of forms retrieved may include searchable forms from many different domains (see e.g., Figure 2(b)). To identify searchable forms that belong to a given domain,

Domain	Recall	Precision	Accuracy
Airfare	0.91	0.91	0.98
Auto	0.87	0.87	0.93
Book	0.9	0.92	0.96
Hotel	0.96	0.97	0.95
Job	0.87	0.95	0.95

Table 1: Effectiveness of classifier composition.

as the second step of our form-filtering process, we use a more specialized classifier, the *Domain-Specific Form Classifier* (*DSFC*). The DSFC uses the textual content of a form to determine its domain. Intuitively, the form content is often a good indicator of the database domain—it contains metadata and data that pertain to the database. For example, form attribute names often match the names of fields in the database, and selection lists often contain values that are present in the database.

The results of preliminary experiments over representative database domains, shown in Figure 1, indicate that the combination of the GFC and DSFC is very effective and lead to high values for classification recall, precision and accuracy.

4. DISCUSSION

This paper presents the first end-to-end solution to the problem of automatically constructing topic-specific online database collections. It combines a focused crawler, that automatically and efficiently locates forms on the Web, with a form-filtering process that accurately identifies forms that belong to a particular database domain.

Unlike previous approaches to form classification which require manual pre-processing of forms (see e.g., [5]), all the features used in our form-filtering process can be automatically extracted from Web pages. Besides, by partitioning the feature space, not only can simpler classifiers be constructed that are more accurate and robust, but this also enables the use of learning techniques that are more effective for each feature subset. Our experiments show that this composition is effective and outperforms solutions based on monolithic classifiers that consider the whole feature space.

5. REFERENCES

[1] L. Barbosa and J. Freire. Searching for Hidden-Web Databases. In *Proceedings of WebDB*, pages 1–6, 2005.

[2] L. Barbosa and J. Freire. Combining classifiers to identify online databases. Technical report, University of Utah, 2006.

[3] S. Chakrabarti, M. van den Berg, and B. Dom. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. *Computer Networks*, 31(11-16):1623–1640, 1999.

[4] M. Galperin. The molecular biology database collection: 2005 update. *Nucleic Acids Res*, 33, 2005.

[5] A. Hess and N. Kushmerick. Automatically attaching semantic metadata to web services. In *Proceedings of IIWeb*, pages 111–116, 2003.

[6] W. Hsieh, J. Madhavan, and R. Pike. Data management projects at Google. In *Proceedings of ACM SIGMOD*, pages 725–726, 2006.

[7] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[8] W. Wu, C. Yu, A. Doan, and W. Meng. An Interactive Clustering-based Approach to Integrating Source Query interfaces on the Deep Web. In *Proceedings of ACM SIGMOD*, pages 95–106, 2004.