International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Focused and Adaptive Crawling for Topic Specific and Hidden Web Entries

Vrutuja Pande[1], Pratap Singh[2]

[1]Pune University, Maharashtra, India

[2]Professor, Pune University, Maharashtra, India

Abstract: ***In this paper we describe new adaptive crawling strategies to efficiently locate the entry points to hidden-Web sources and we describe a new hypertext resource discovery system called a Focused Crawler. The fact that hidden-Web sources are very sparsely distributed makes the problem of locating them especially challenging. We deal with this problem by using the contents of pages to focus the crawl on a topic; by prioritizing promising links within the topic; and by also following links that may not lead to immediate benefit. We propose a new framework whereby crawlers automatically learn patterns of promising links and adapt their focus as the crawl progresses, thus greatly reducing the amount of required manual setup and tuning. The goal of a focused crawler is to selectively seek out pages that are relevant to a pre-defined set of topics. The topics are specified not using keywords, but using exemplary documents. Rather than collecting and indexing all accessible Web documents to be able to answer all possible ad-hoc queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoid and network resources, and helps keep the crawl more up-to-dates we designed two hypertext mining programs that guide our crawler: a classifier that evaluates the relevance of a hypertext document with respect to the focus topics, and a distiller that identifies hypertext nodes that are great access points to many relevant pages within a few links, Irrelevant regions of the Web. This leads to significant savings in hardware. Our experiments over real Web pages in a representative set of domains indicate that online learning leads to significant gains in harvest rates—the adaptive crawlers retrieve up to three times as many forms as crawlers that use a fixed focus strategy.***

Keywords: Web resource discovery; Classification; Categorization, Web crawling strategies

1. Introduction

The hidden Web has been growing at a very fast pace. It is estimated that there are several million hidden-Web sites. These are sites whose contents typically reside in databases and are only exposed on demand, as users fill out and submit forms. As the volume of hidden information grows, there has been increased interest in techniques thatallow users and applications to leverage this information. Examples of applications that attempt to make hidden-Web information more easily accessible include: meta Searchers, hidden-Web crawlers, online-database directories and Web information integration systems. Since for any given domain of interest, There are many hidden-Web sources whose data need to be integrated or searched; a key requirement for these applications is the ability to locate these sources. But doing so at a large scale is a challenging problem. The crawler must also produce high-quality results. Havinga homogeneous set of forms that lead to databases in thesame domain is useful, and sometimes required, for a numberof applications. For example, the effectiveness of form integration techniques can be greatly diminished if the set of input forms is noisy and contains forms that are not in the integration domain. However, an automated crawling process invariably retrieves a diverse set of forms. A focus topic may encompass pages that contain searchable forms from many different database domains. For example, while Crawling to find Airfare search interfaces a crawler is likely to retrieve a large number of forms in different domains, such as Rental Cars and Hotels, since these are often co-located with Airfare search interfaces in travel sites. The set of retrievedforms also includes many non-searchable forms thatdo not represent database queries such as forms for login,mailing list subscriptions, quote requests, and Web-basedemail forms. The Form-Focused Crawler (FFC) was our first attempt to address the problem of automatically locating online databases. The FFC combines techniques for focusing the crawl on a topic with a link classifier which identifies and prioritizes links that are likely to lead to searchable forms in one or more steps. Our preliminary results showed that the FFC is up to an order of magnitude more efficient, with respect to the number of searchable forms it retrieves, than a crawler that focuses the search on topic only. This approach, however, has important limitations. First, it requires substantial manual tuning, including the selection of appropriate features and the creation of the link classifier. In addition, the results obtained are highly-dependent on the quality of the set of forms used as the training for the link classifier. If this set is not representative, the crawler may drift away from its target and obtain low harvest rates.

Given the size of the Web, and the wide variation in the hyperlink structure, manually selecting a set of forms that cover a representative set of link patterns can be challenging. Last, but not least, the set of forms retrieved by the FFC is very heterogeneous—it includes all searchable forms found during the crawl, and these forms may belong to distinct database domains

2. Background: The Form Focused Crawler

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. The main components of the FFC are shown in white in Figure 1 and are briefly described below.• The page classifier is trained to classify pages as belonging to topics in a

taxonomy (e.g., arts, movies, jobs in Dmoz). It uses the same strategy as the best-first crawler of once the crawler retrieves a page P, if P is classified as being On-topic, its forms and links are extracted. • The link classifier is trained to identify links that are likely to lead to pages that contain searchable form interfaces in one or more steps. It examines links extracted from on-topic pages and adds the links to the crawling frontier in the order of their predicted reward. • The frontier manager maintains a set of priority queues with links that are yet to be visited. At each crawling step, it selects the link with the highest priority.• The searchable form classifier filters out non-searchable forms and ensures only searchable forms are added to the Form Database. This classifier is domain-independent and able to identify searchable forms with high accuracy. The crawler also employs stopping criteria to deal with the fact that sites, in general, contain few searchable forms. It leaves a site after retrieving a pre-defined number of distinct forms, or after it visits a pre-defined number of pages in the site.

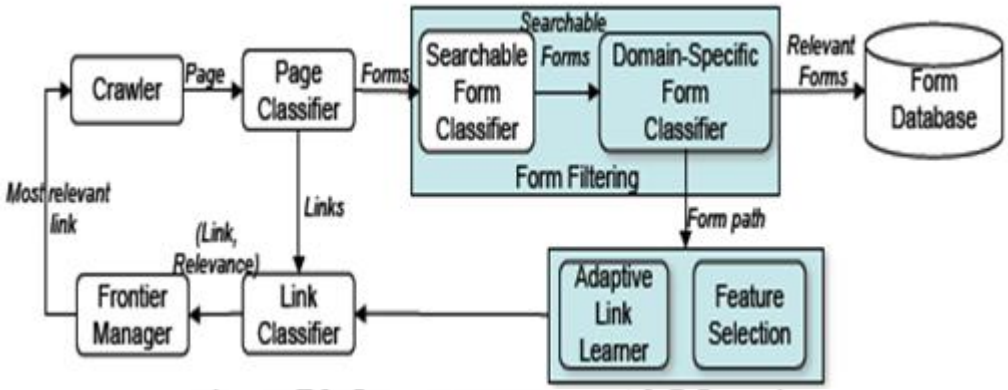

Figure 1: Architecture of ACHE the new modules that are responsible for the online focus adaptation are shown in blue; and the modules shown in white are used both in the FFC and in ACHE

3. Limitation of FFC

An experimental evaluation of the FFC [3] showed that FFC is more efficient and retrieves up to an order of magnitude more searchable forms than a crawler that focuses only on topic. In addition, an FFC configuration with a link classifier that uses multiple levels performs uniformly better than their counterpart with a single level (i.e., a crawler that focuses only on immediate benefit). The improvements in harvest rate for the multi-level configurations varied between 20% and 110% for the three domains we considered. This confirms results obtained in other works which underline the importance of taking delayed benefit into account for sparse concepts.

The strategy used by the FFC has two important limitations. The set of forms retrieved by the FFC is highly heterogeneous. Although the Searchable Form Classifier is able to filter out non-searchable forms with high accuracy, a qualitative analysis of the searchable forms retrieved by the FFC showed that the set contains forms that belong to many different database domains. The average percentage of relevant forms (i.e., forms that belong to the target domain) in the set was low—around 16%. For some domains the percentage was as low as 6.5%. Whereas it is desirable to list only relevant forms in online database directories, such as Bright Planet and the Molecular Biology Database Collection, for some applications this is a requirement. Having a homogeneous set of the forms that belong to the same database domain is critical for techniques such as statistical schema matching across Web interfaces, whose effectiveness can be greatly diminished if the set of input forms is noisy and contains forms from multiple domains. Another limitation of the FFC is that tuning the crawler and training the link classifier can be time consuming. The process used to select the link classifier features is manual: terms deemed as representative are manually selected for each level. The quality of these terms is highly-dependent on knowledge of the domain and on whether the set of paths obtained in the back-crawl is representative of a wider segment of the Web for that database domain. If the link classifier is not built with a representative set of paths for a given database domain, because the FFC uses a fixed focus strategy, the crawler will be confined to a possibly small subset of the promising links in the domain.

4. Our contribution: The Crawler Focus

With the goal of further improving crawler efficiency, the quality of its results, and automating the process of crawler setup and tuning, we use a learning-agent-based approach to the problem of locating hidden-Web entry points. Learning agents have four components

- The behavior generating element (BGE), which based on the current state, selects an action that tries to maximize the expected reward taking into account its goals (exploitation);• Theproblem generator (PG) that is responsible for suggesting actions that will lead to new experiences, even if the benefit is not immediate, i.e., the decision is locally suboptimal (exploration);
- The critic that gives the online learning element feedback on the success (or failure) of its actions; and
- The online learning element which takes the critic's feedback into account to update the policy used by the BGE.

4.1 The ACHE Architecture

In ACHE, we employ the adaptive link learner as the learning element. It dynamically learns features automatically extracted from successful paths by the feature selection component, and updates the link classifier. The effectiveness of the adaptive link learner depends on the accuracy of the form-filtering process; on the ability of the

feature selector to identify 'good' features; and on the efficacy of the frontier manager in balancing exploration and exploitation. Below we describe the components and algorithms responsible for making ACHE adaptive.

4.2 Adaptive Link Learner

The adaptive link learner, in contrast, uses features of paths that are gathered during the crawl. ACHE keeps a repository of successful paths: when it identifies a relevant form, it adds the path it followed to that form to the repository. The adaptive link learner is invoked periodically, when the learning threshold is reached (line 1). For example, after the crawler visits a pre-determined number of pages, or after it is able to retrieve a pre-defined number of relevant forms. Note that if the threshold is too low, the crawler may not be able to retrieve enough new samples to learn effectively. On the other hand, if the value is too high, the learning rate will be slow. In our experiments, learning iterations are triggered after 100 new relevant forms are found

4.3 Automating the Feature Selection Process

The Automatic Feature Selection (AFS) algorithm extracts features present in the anchor, URL, and text around links that belong to paths which lead to relevant forms. The feature selection process must produce features that are suitable for the learning scheme used by the underlying classifier. Initially, all terms in anchors are extracted to construct the anchor feature set. For the around feature set, AFS selects the n terms that occur before and the n terms that occur after the anchor (in textual order). Because the number of extracted terms in these different contexts tends to be large, stop-words are removed and the remaining terms are stemmed. The most frequent terms are then selected to construct the feature set .The URL feature space requires special handling. Since there is little structure in a URL, extracting terms from a URL is more challenging. For example, ―job search" and ―used cars" are terms that appear in URLs of the Job and Auto domains, respectively. To deal with this problem, we try to identify meaningful sub-terms using the following strategy. After the terms are stemmed, the k most frequent terms are selected. Then, if a term in this set appears as a substring of another term in the URL feature set, its frequency is incremented. Once this process finishes, the k most frequent terms are selected.

4.4 Form Filtering

The form filtering component acts as a critic and is responsible for identifying relevant forms gathered by ACHE.It assists ACHE in obtaining high-quality results and it also enables the crawler to adaptively update its focus strategy, as it identifies new paths to relevant forms during a crawl. Therefore, the overall performance of the crawler agent is highly-dependent on the accuracy of the form-filtering process. If the classifiers are inaccurate, crawler efficiency can be greatly reduced as it drifts way from its objective through unproductive paths. The form filtering process needs to identify, among the set of forms retrieved by the crawler, forms that belong to the target database domain. Even a focused crawler retrieves a highly-heterogeneous set of forms. A focus topic (or concept) may encompass pages that contain many different database domains. For example, while crawling to find airfare search interfaces the FFC also retrieves a large number of forms for rental car and hotel reservation, since these are often co located with airfare search interfaces in travel sites. The retrieved forms also include non searchable forms that do not represent database queries such as forms for login, mailing list subscriptions, and Web-based email form

5. System Architecture

The focused crawler has three main components:a **classifier** which makes relevance judgments onpages crawled to decide on link expansion, a **distiller**which determines a measure of centrality ofcrawled pages to determine visit priorities, and a**crawler** with dynamically reconfigurable priority controls which is governed by the classifier anddistiller. A block diagram is shown in Fig. 2

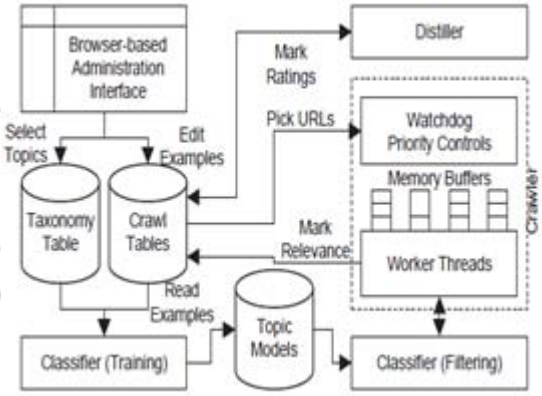

Figure 2: Block diagram of the focused crawler showing how the crawler, classifier and distiller are integrated

6. Conclusion and Future Work

We have presented a new adaptive focused crawling strategy for efficiently locating hidden-Web entry points. This strategy effectively balances the exploitation of acquired knowledge with the exploration of links with previously unknown patterns, making it robust and able to correct biases introduced in the learning process. this framework can greatly reduce the effort to configure a crawler. In addition, by using the form classifier, ACHE produces high quality results that are crucial for a number information integration tasks. To accelerate the learning process and better handle very sparse domains, we will investigate the effectiveness and trade-offs involved in using back-crawling during the learning iterations to increase the number of sample paths. The focused crawler is a system that learns the specialization from examples, and then explores the Web; our system selects work very carefully from the crawl frontier. A consequence of the resulting efficiency is that it is feasible to crawl to a greater depth than would otherwise be possible.

References

[1] S. Chakrabarti, D. Gibson and K. McCurley, Surfing theweb backwards, in: 8th World Wide Web Conference,Toronto, Canada, May 1999.

[2] L. Barbosa and J. Freire. Combining classifiers toidentify online databases.In Proceedings of WWW,2007.
[3] Yang and J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. InInternational Conference on Machine Learning, pages412–420, 1997.
[4] C. C. Aggarwal, F. Al-Garawi, and P. S. Yu.Intelligent crawling on the world wide web witharbitrary predicates. In Proceedings of WWW, pages96–105, 2001.
[5] S. Chakrabarti, B. Dom, R. Agrawal and P. Raghavan,Scalable feature selection, classification and signature generationfor organizing large text databases into hierarchicaltopic taxonomies, VLDB Journal 7(3): 163–178, 1998.