Intelligent Web Crawling

(WI-IAT 2013 Tutorial)

Denis Shestakov

Abstract—**Web crawling, a process of collecting web pages in an automated manner, is the primary and ubiquitous operation used by a large number of web systems and agents starting from a simple program for website backup to a major web search engine. Due to an astronomical amount of data already published on the Web and ongoing exponential growth of web content, any party that want to take advantage of massive-scale web data faces a high barrier to entry. We start with background on web crawling and the structure of the Web. We then discuss different crawling strategies and describe adaptive web crawling techniques leading to better overall crawl performance. We finally overview some of the challenges in web crawling by presenting such topics as collaborative web crawling, crawling the deep Web and crawling multimedia content. Our goals are to introduce the intelligent systems community to the challenges in web crawling research, present intelligent web crawling approaches, and engage researchers and practitioners for open issues and research problems. Our presentation could be of interest to web intelligence and intelligent agent technology communities as it particularly focuses on the usage of intelligent/adaptive techniques in the web crawling domain.**

***Index Terms*—web crawling, web crawler, intelligent crawling, adaptive crawling, collaborative crawling, Web ecosystem, Web structure, incremental crawling, focused crawling, deep Web**

I. Introduction

WEB crawling [1], [2], a process of collecting web pages in an automated manner, is the primary and ubiquitous operation used by a large number of web systems and agents starting from a simple program for website backup to a major web search engine. For example, search engines such as Google or Microsoft Bing use web crawlers to routinely visit billions of web pages, which are then indexed and made available for answering user search requests. In this way, the characteristics of obtained web crawls such as coverage or freshness directly affect on the quality of web search results served to users. Besides web search, the web crawling technology is central in such applications as web data mining and extraction, web monitoring, social media analysis, digital preservation (i.e., web archiving), detection of web spam and fraudulent web sites, web application testing, finding unauthorized use of copyrighted content (music, videos, texts, etc.), identification of illegal and harmful web activities (e.g., terrorist chat rooms), and virtual tourism.

Due to an astronomical amount of data already published on the Web and ongoing exponential growth of web content, any party (be it an individual, company, government agency, nonprofit or educational organization) that want to take advantage of massive-scale web data faces a high barrier to entry. Indeed, only network costs associated with the downloading of web-scale size collection by themselves lead to expenses that are not affordable by the majority of potential players.

For those with flexible budgets, there is a next barrier: operating web-scale crawl, i.e. hundreds of millions of pages, is a challenging task that requires skills and expertise in distributed data retrieval and processing, not to mention large operational costs. Finally, for the parties who nevertheless manage to overcome the above obstacles but interested in specific subsets of web information, the results of crawl are often wasteful, as majority of retrieved pages do not match their criteria of interest.

In this paper, we will overview recent advances made in harvesting the information on the Web, in order to introduce the intelligent systems community to the challenges in this area, with particular stress on intelligent web crawling approaches using adaptive crawling agents as well as the underlying open issues and research problems. We will also address issues in building a spectrum of services and applications collecting and aggregating large amounts of web information, e.g., the role of web crawlers in the Web ecosystem, how intelligent crawling strategies can lead to a better overall quality of crawled data.

II. Web Crawling

This section will introduce the basics of web crawler operations and important web crawling applications, and provide relevant statistics on the Web link structure. Next we will describe the architecture of a web crawler and present a number of crawling strategies including three adaptive crawling approaches.

A. Overview

The underlying mechanism of crawling – namely, given an URL download a corresponding web page, extract all URL links from it and repeat the process for those links that were not visited yet – is naive and simple. However, due to a number of imposing restrictions and resource limitations under which crawlers operate, algorithms and techniques behind a large-scale web crawler are far more complicated than the trivial implementation. For example, in order not to be banned by a web server, a crawler has to avoid sending too many URL requests to a server within a short time period. Since the distribution of pages over web servers is non-uniform, a crawler faces a problem of downloading a large number of pages from only a relatively small number of web servers (comparing to their overall number on the Web).

D. Shestakov is with the Department of Media Technology, Aalto University, Finland e-mail: denis.shestakov@aalto.fi

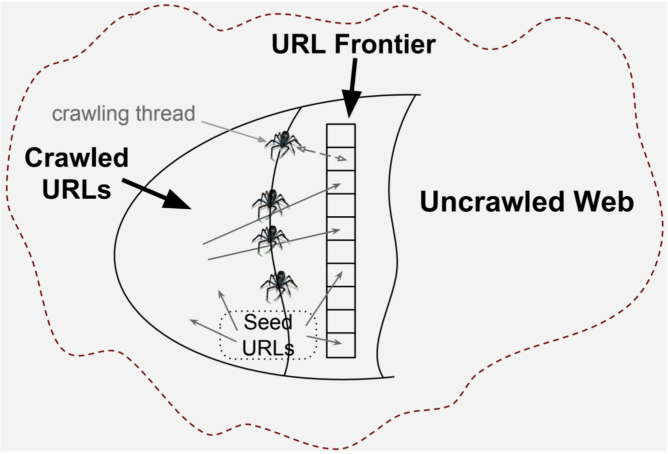

Fig. 1. URL Frontier in crawler's operations.

There are many applications with web crawlers playing a crucial role. The application spectrum ranges from visiting as many web pages as possible by web search engine or web archiver crawlers to the recently appeared trend of using crawlers for web application testing [3]. Needs of commercial web search engines are, however, the most important driving force in design and development of better crawler agents. With a few notable exceptions (e.g., see [4], [5]), academic crawling projects operate on a much smaller scale and apparently employ less sophisticated techniques.

The size and structure of the Web [6] are the most essential aspects that define several key requirements for a web crawler. The exponential growth of the Web suggests that no crawler can cope to cover all the information on the public Web [7], [8]. Moreover, the dynamism of web content guarantees that any collection of crawled documents is stale (not up-to-date) to a certain degree. As normally only limited resources are available, making crawls to be up-to-date involves a trade-off between freshness and coverage of the harvested documents. Similarly, the link structure of the Web [9] is crucial for understanding how crawlers can better prioritize their unseen URL lists.

B. Intelligent Web Crawling

A general-purpose web crawler typically operates in a distributed fashion, with multiple crawl threads that may run under different processes and often at different nodes. The architecture of a crawler [10], [11] includes a number of components, including the URL frontier. It keeps URLs to be visited in some order and returns the one with the highest score to a crawler thread when it seeks for a new URL. The URL frontier is schematically depicted in Figure 1.

There are a number of approaches to prioritize the URLs in the URL frontier. The main goal is to assign a URL some value that corresponds to the "importance" of a web page located at this URL. The URL prioritization strategies clearly depend on the crawling goals. E.g., if a crawler has no domain focus (general) or has to primarily focus on harvesting pages on a certain topic (topical). Another possible concern could be if a crawler should make a snapshot of a certain segment of the Web (batch) or should re-crawl previously visited pages (incremental). In general, one can categorize the existing approaches into six popular strategies used for both general and topical batch crawling: Breadth-First, Depth-First, Backlink count, Best-First, PageRank and Shark-Search [12], [13]. In essence, a crawling strategy defines the assignment of a priority value to a newly extracted URL. Depending on the strategy a number of factors can be taken into account – from a simple time-stamp of adding a link to the frontier to an inherited score value based on relevance scores of several ancestor pages pointing to a page with this link.

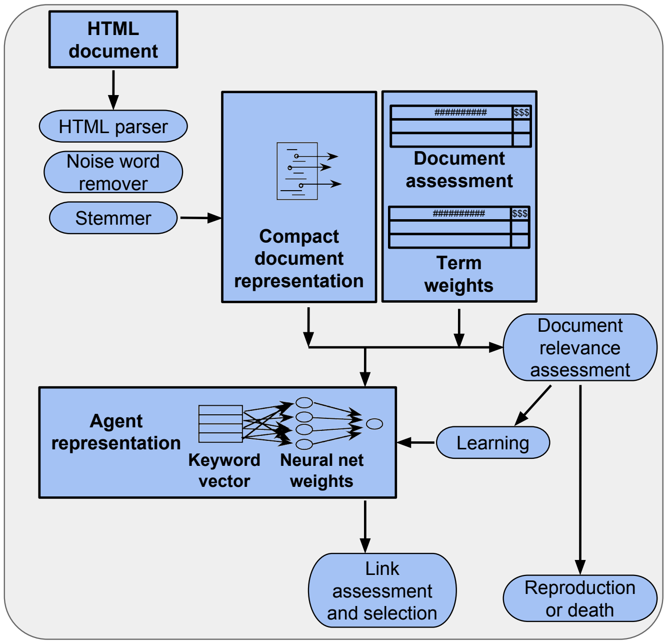

Fig. 2. Architecture of InfoSpider agent.

The abovementioned crawling strategies are static, in the sense that they do not learn from experience or adapt to the context of a topic in the course of crawl. In contrast to them, an intelligent crawler agent uses an adaptive learning model to assign priorities to the URLs in the frontier. In the literature, there exist at least three adaptive crawling approaches: InfoSpiders, ant-based crawling and HMM-supported crawling [14], [15], [16], [17]. While HMM-supported crawling utilizes Hidden Markov Models for learning paths leading to relevant pages, InfoSpiders and ant-based crawling are inspired by evolutionary biology studies and models of social insect collective behaviour correspondingly. Figure 2 shows the architecture of the InfoSpider agent, where agent's representation is supported by neural network.

III. Open Challenges

This section will briefly discuss the role of crawlers in the Web ecosystem and then present some open challenges in web crawling research, such as collaborative web crawling, crawling the deep Web and crawling multimedia content.

Being an important part of the Web ecosystem, crawler agents follow the pull model of resource access, under which a client has to first issue a request for a given resource (compared with the push model where a server can send (push) a content to a client without an earlier request from client-side). While the pull model has several advantages, it also leads to significant inefficiencies in crawlers' performance. The collaborative crawling or "crawling as a common service" approach [18] is the attempt to overcome some of these problems by supplementing a regular general crawler with a scalable filtering layer that allows other parties to crawl by setting conditions for documents of interest and obtaining relevant documents from the prime crawler.

The significant portion of the Web containing publicly-available information from myriads of online web databases (known as the deep Web [19]) is poorly accessible by crawlers. Accessing a deep web resource requires recognizing a search interface (search form) to a database and filling the recognized interface with meaningful values – both tasks are extremely challenging for conventional crawlers. In the literature, there are some relevant techniques for deep web crawling [20], [21], [22].

The Web has evolved from a huge textual repository to a fully-fledged multimedia platform serving web users all media types of content. Images, video, audio are now not just supplementing textual content of web documents but become integral part of many web resources. Most crawlers, however, do not adapt to this change and continue to operate as text harvesting systems. Thus, problems in crawling multimedia content [23], [24] are well-timed and of high importance.

IV. Supporting Materials

The material of this article was presented as a tutorial on the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT 2013) held in Atlanta, USA in November 2013. The tutorial slides are available at http://goo.gl/woVtQk; note that last part of tutorial provides relevant references to important crawl datasets and self-study materials. The bibliography for web crawling domain can be found in [1], [25].

References

[1] C. Olston and M. Najork, "Web crawling," *Found. Trends Inf. Retr.*, vol. 4, no. 3, pp. 175–246, 2010.

[2] D. Shestakov, "Current challenges in web crawling," in *Proc. ICWE 2013*, 2013, pp. 518–521.

[3] S. Choudhary, M. E. Dincturk, S. M. Mirtaheri, A. Moosavi, G. von Bochmann, G.-V. Jourdan, and I. V. Onut, "Crawling rich internet applications: The state of the art," in *Proc. CASCON 2012*, 2012, pp. 146–160.

[4] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov, "IRLbot: Scaling to 6 billion pages and beyond," *ACM Trans. Web*, vol. 3, no. 3, pp. 8:1–8:34, 2009.

[5] P. Boldi, A. Marino, M. Santini, and S. Vigna, "Bubing: Massive crawling for the masses," 2013. [Online]. Available: http://vigna.di.unimi.it/papers/P4.8.pdf

[6] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "The web as a graph: Measurements, models, and methods," in *Proc. COCOON 1999*, 1999, pp. 1–17.

[7] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index," *J. ACM*, vol. 55, no. 5, pp. 24:1–24:74, 2008.

[8] D. Shestakov, "Sampling the national deep web," in *Proc. DEXA 2011*, 2011, pp. 331–340.

[9] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, and E. Upfal, "The web as a graph," in *Proc. PODS 2000*, 2000, pp. 1–10.

[10] V. Shkapenyuk and T. Suel, "Design and implementation of a high-performance distributed web crawler," in *Proc. ICDE 2002*, 2002, pp. 357–368.

[11] P. Boldi, B. Codenotti, M. Santini, and S. Vigna, "Ubicrawler: a scalable fully distributed web crawler," *Software: Practice and Experience*, vol. 34, no. 8, pp. 711–726, 2004.

[12] M. Najork and J. L. Wiener, "Breadth-first crawling yields high-quality pages," in *Proc. WWW 2001*, 2001, pp. 114–118.

[13] M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, "The shark-search algorithm. an application: tailored web site mapping," *Computer Networks and {ISDN} Systems*, vol. 30, no. 1-7, pp. 317–326, 1998.

[14] F. Menczer, G. Pant, and P. Srinivasan, "Topical web crawlers: Evaluating adaptive algorithms," *ACM Trans. Internet Technol.*, vol. 4, no. 4, pp. 378–419, 2004.

[15] A. Micarelli and F. Gasparetti, "Adaptive focused crawling," in *The Adaptive Web*, ser. Lecture Notes in Computer Science, 2007, vol. 4321, pp. 231–262.

[16] F. Gasparetti and A. Micarelli, "Swarm intelligence: Agents for adaptive web search," in *Proc. ECAI 2004*, 2004, pp. 1019–1020.

[17] S. Batsakis, E. G. Petrakis, and E. Milios, "Improving the performance of focused web crawlers," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 1001–1013, 2009.

[18] J. M. Hsieh, S. D. Gribble, and H. M. Levy, "The architecture and implementation of an extensible web crawler." in *Proc. NSDI 2010*, 2010, pp. 329–344.

[19] D. Shestakov, "Deep web: databases on the web," in *Encyclopedia of Database Technologies and Applications, 2nd edition*, 2009, pp. 581–588.

[20] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, "Google's deep web crawl," *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1241–1252, 2008.

[21] D. Shestakov, "On building a search interface discovery system," in *Proc. VLDB Workshops 2009*, 2010, pp. 81–93.

[22] Z. Behfarshad and A. Mesbah, "Hidden-web induced by client-side scripting: an empirical study," in *Proc. ICWE 2013*, 2013, pp. 52–67.

[23] A. Nanopoulos, D. Rafailidis, M. M. Ruxanda, and Y. Manolopoulos, "Music search engines: Specifications and challenges," *Information Processing & Management*, vol. 45, no. 3, pp. 392–396, 2009.

[24] J. Hare, D. Dupplaw, W. Hall, P. H. Lewis, and K. Martinez, "Building a multimedia web observatory platform," 2013. [Online]. Available: http://eprints.soton.ac.uk/352465/

[25] (2013) Web Crawling (Mendeley Group). [Online]. Available: http://www.mendeley.com/groups/531771/web-crawling/