Web Crawler

Pant et al. [1] define **Web Crawlers** as programs that exploit the graph structure of the Web to move from page to page. Crawlers are also known as spiders, robots or simply bots, and in their infancy were also named wanderers, fish or worms. In 1993, the first web crawler, the World Wide Web Wanderer[1], was used to compile statistics about the growth of the web [2]. They quickly became a central component of search engines to collect web pages that are to be indexed (for example the googlebot[2]). Crawlers are used for many other applications such as archiving the web (e.g. Heritrix[3] crawler) or specialized data mining, for example for price comparison services (e.g. the ShopWiki[4] crawler) or searching for copyright violation (e.g. the Digimarc Guardian[5] or the Belgian Librius SINBAD[6] crawlers) and more (e.g. spambot (used to collect email addresses and send unsolicited messages), etc.).

The operations of web crawlers, as described by Manning et al. [3, pp. 405-420], are as follow:

1. The crawler starts with one or more web page addresses constituting the *seed set*, picks one URL[7] from the set and download that web page.
2. The fetched page is then parsed with two objectives:
   (a) Extract the text, images, etc. for the data mining process, or in the case of a search engine to fed the indexer.
   (b) Extract the links to other pages, files, etc. that are added to a *URL frontier*, corresponding to the resources that have yet to be fetched by the crawler. Initially, the URL frontier contains the seed set.
3. Once the resource has been fetched and parsed, its URL is either removed from the URL frontier or time stamped for a later visit.

Manning et al. [3, pp. 405-420] also list the features web crawler must or should provide. Ideally, a crawler must be **robust** to avoid spider trap, which are server generating web pages making the crawler downloading million of pages from the same domain and be **polite**. The politeness is both implicit, for example the crawler will wait some time before downloading a page from the same domain to avoid server overloading and bandwidth consumption and explicit by following the Robots Exclusion Protocol define in the robot.txt[8] which set what resources a crawler can or not visit and fetch. Malware and spam robots are know to not follow these rules and some commercial crawler such as the Attributor (now Digimarc Guardian) do not respect the exclusion neither, even if understandably necessary, webmasters may block it more aggressively[9]. The features crawler should have concern efficiency, quality, freshness, scalability, extensibility, etc.

[1] http://www.mit.edu/people/mkgray/net/background.html

[2] http://support.google.com/webmasters/bin/answer.py?hl=en&answer=182072

[3] https://webarchive.jira.com/wiki/display/Heritrix/Heritrix

[4] http://www.shopwiki.com/w/Help:Bot

[5] http://www.attributor.com/solutions/solutions.php?X=1.1

[6] http://www.librius.com/over-librius/wat-doet-librius/Librius-SINBAD/

[7] Uniform Resource Locator

[8] http://www.robotstxt.org/

[9] http://incredibill.blogspot.fi/2007/11/attributor-post-mortem-copyright.html

References

[1] Gautam Pant, Padmini Srinivasan, Filippo Menczer. Crawling the Web. In: M Levene, A Poulovassilis, editors. In Web Dynamics: Adapting to Change in Content, Size, Topology and Use. Springer-Verlag; 2004. p. 153--178.

[2] Marc Najork. Web Crawler Architecture. In: Ling Liu, M Tamer Özsu, editors. Encyclopedia of Database Systems. Springer US; 2009. p. 3462--3465.

[3] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze. Introduction to Information Retrieval. Cambridge University Press; 2008.