Chapter 1

Conclusions

During this thesis, We studied Web crawling at many different levels. Our main objectives were to develop a model for Web crawling, to study crawling strategies and to build a Web crawler implementing them. This section points out what we have done and what could be done as future work.

The next section summarizes our contributions, Section 1.2 describes some guidelines for future work, and Section 1.3 discusses open problems for Web crawling.

1.1 Summary of our contributions

This thesis dealt with Web crawling from a practical point of view. From this point of view, we worked in a series of problems which appeared during the design and implementation of a Web crawler.

We started by describing the topic of Web crawling and summarizing relevant literature on the topic. Although there are many studies about Web search, Web crawler designs and algorithms are mostly kept as business secret, with some exceptions. (Chapter **??**).

Our model divides the problem of Web crawling into short-term and long-term scheduling. We explained why this separation is possible and how we can exploit it for efficient Web crawling, by generating batches of pages using a long-term scheduling policy, and then re-ordering these batches to apply a short-term scheduling policy. The long-term scheduling is related to page quality, while the short-term scheduling is related to network efficiency (Chapter **??**).

We also proposed a better integration between the Web crawler and the rest of the search engine, and a framework for measuring the quality of a crawl. Within this framework, we classified several existing Web crawlers according to the relevance they give to different parameters of our model (Chapter **??**).

Regarding page quality, we compared several scheduling algorithms for long-term scheduling, and we found a very simple and fast strategy for downloading important pages early in the crawl. We ran experiments

in both a simulated and a real Web environment for proving that this strategy is successfull. For short-term scheduling –network efficiency– we showed the effect of downloading several pages using the same connection and explained why a long waiting time between accesses to pages in the same server decreases crawling efficiency (Chapter **??**).

It become clearly very quickly that if the crawler downloads dynamic pages then the space of URLs to download from is practically infinite. In this context, it is important to know when to stop a crawl. Instead of setting and arbitrary crawling depth, we studied this problem by estimating the estimate the value of the pages that were not crawled. To do this, user behavior was studied and modeled using a Markov chain created from data from actual Web sites. The conclusion was that users explorations are in general very shallow, and we were able to set an appropiate crawling depth (Chapter **??**).

To improve freshness in the search engine, there are a number of actions that Web server administrators can take. We proposed and compared several cooperation schemes for Web servers and Web crawlers, including polling and interrupt versions for them. These cooperation schemes can help a Web crawler in detecting changes in Web sites, and lower the network usage of the crawling process, which is benefical for both the search engine and the Web site (Chapter **??**).

We implemented a high-performance Web crawler in C/C++. This crawler implements the scheduling algorithms proposed in this thesis, and serves as a proof of concept for the cooperation schemes. Our implementation is publicly available as it is released under the GNU public license. This Web crawler can efficiently download several million pages per day, and can be used for Web search and Web characterization (Chapter **??**).

We downloaded and analyzed the Chilean Web using our crawler, extracting statistics on HTML pages, multimedia files, Web sites and link structure. We also used the crawler for downloading the Greek Web and compared both datasets, finding that despite large differences in the context of both countries –for instance, the GDP per capita of Greece doubles the one of Chile, as well as the number of pages– both Web collections are very similar. (Chapter **??**).

During the crawling processes we carried, we discovered and studied several practical issues that arise only after large crawls. Those issues are mostly anomalies in the implementations of the underlying protocols that form the Web, and must be taken into account when developing a Web crawler (Appendix **??**).

1.2 Future work

As most research, this thesis opens more problems than it solves. Reviewers of our work and ourselves found several avenues for future work. We consider the following as the main ones:

Validating the collection The Chilean Web with its 3.5 million pages represents 1/1000 of the Web. We have been studying this collection since 2000, and the distribution of several variables seems very

similar to the data published for other samples of the Web, but this should be tested with other collections.

A related question would be if the Web is a representative collection of text. The answer given by Kilgariff and Grefenstette [KG04] is very pragmatical:

> "The Web is not representative of anything else. But neither are other corpora, in any well-understood sense."

Considering the contents of Web pages Besides checking for duplicates, we largely ignore the text of pages. This is done on purpose: we wanted to study only links to use them as independent information, which can be combined later with other evidence.

In particular, we think that the methods of focused crawling [CvD99], in which a crawler is directed to pages on *specific* topics based on properties of the text, can be adapted to direct the crawler to interesting pages on *all* topics.

Developing complex models of user behavior The models of user behavior we presented in Chapter **??** were chosen for their simplicity. A more complex model would be a model with memory, in which the probability of users following a link depends on their previous actions. We have observed that this is indeed the case, because users that go deeper into the Web site have a higher probability of continuing to follow links. We also pointed out that user behavior in Blogs is different than on other sites.

Testing short-term scheduling strategies We have focused mostly on long-term scheduling strategies, although we made experiments on short-term scheduling and described them in Chapter **??**. The most important issue would be to implement persistent connections in the Web crawler, to download several pages from the Web site without re-connecting. In theory, this has a significant positive impact on Web crawl, but further testing is required.

Continuous crawling We focused mostly in a "crawl-and-stop" type of crawl, in which we only crawl each page once. Our crawler can also continuously check pages for updates, using the value function to decide when to check for a page for changes, and when to download new pages. In this context, the measure of importance is not the time it takes to complete the crawl, but the average freshness and/or the average quality when a stationary state is reached. This is also related to measure how good are the change prediction in a real Web environment.

Benchmarking Measuring the efficiency of a Web crawler –considering only short-term scheduling –is not an easy task. We need a framework for comparing crawlers that accounts for the network usage, the processing power required, and the memory usage. It would be good to have a measure that allows us to compare, e.g., 30 pages per second in a 1Ghz processor with 1Gb RAM with 20 pages per second in a 800MHz processor with 640Mb RAM. A more important problem is that network conditions vary so for testing different Web crawlers we must consider the time of the day, network capacity, etc.

Finding combined strategies Specifically the best parameters for the manager program, in terms of the importance that should be given to intrinsic quality, representational quality and freshness. We also consider that the scheduling policies should adapt to different Web sites, and our model provides a framework for that kind of adaptability.

Exploiting query logs for Web crawling The usage of user query logs in a search engine for guiding a Web crawling is an important step in integrating the collection and search processes. The query logs can be used to refresh frequently returned pages faster, so the crawler refreshes the active set of the search engine more often. Moreover, the query terms used in Web search can be given priority, so the Web crawler scheduling could be biased toward pages that contains the query terms that are being queried more frequently by the search engine's users.

1.3 Open problems

The importance of a good crawler strategy depends on the balance between these quantities:

1. The total amount of information available on the Web.
2. The amount of bandwidth available for the Web crawler.
3. The amount of good information about specific topics on the Web.

Web search is difficult today because the Web is very large, the bandwidth available at most locations is relatively small, and good pages are also relatively scarce. The three quantities are very dynamic, and their evolution is very difficult to predict. For instance, there is no clear equivalent of a "Moore's law" for network connectivity. Thus, we do not know if the problem of keeping the repository of a search engine is going to get easier or harder.

It is likely that Web crawling continues to be a difficult problem, at least during the next years, and we expect several challenges. Multimedia information such as digital photographies and recordings could account for a larger proportion of the Web content, and the number of Web posting in Blogs will be larger than the number of Web pages, further reducing the signal-to-noise ratio of the Web. Finally, pages with semantic markup could become a significant fraction of Web pages, radically changing the problem of Web search.

Bibliography

[CvD99] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11–16):1623–1640, 1999.

[KG04] Adam Kilgarriff and Gregory Grefenstette. Introduction to the special issue on the Web as corpus. *Computational Linguistics*, 29(3):333–348, 2004.