Computer Standards & Interfaces 43 (2016) 1–11

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A focused crawler combinatory link and content model based on T-Graph principles

Ali Seyfi [a,*], Ahmed Patel [b,c]

[a] Department of Computer Science The George Washington University, Washington, DC, United States
[b] Faculty of Computer Science and Information Systems, Jazan University, Saudi Arabia
[c] Faculty of Science, Engineering and Computing, Kingston University, United Kingdom

ARTICLE INFO

Article history:
Received 24 August 2014
Received in revised form 5 July 2015
Accepted 9 July 2015
Available online 14 July 2015

Keywords:
Focused Web crawler
T-Graph
HTML data
Information retrieval
Search engine

ABSTRACT

The two significant tasks of a focused Web crawler are finding relevant documents and prioritizing them for effective download. For the first task, we propose an algorithm to fetch and analyze the most effective HTML elements of the page to predict and elicit the topical focus of each unvisited page with high accuracy. For the second task, we propose a scoring function of the relevant URLs through the use of T-Graph to prioritize each unvisited link. Thus, our novel method uniquely combines these approaches, giving precision and recall values close to 50%, which indicate the significance of the proposed architecture.

1. Introduction

The openly indexable World Wide Web has empirically passed 60 trillion documents, with more than 50 billion pages indexed daily [1–3] and hitherto, growth does not appear to be leveling off. Considering the exponentially increasing amount of dynamic content on the Web, such as news, schedules, social networks and individual data, it is evidently asserted that relying on search engines is inevitable for the people to approach their desired information. Whereof these concerns make searching the Web a profound task, experts apply machine-learning algorithms to accomplish several phases of this job such as ranking retrieved Web pages based on their estimated relevance to the user query. The main goal is to make up the best weighting algorithm that could represent queries and pages as in a vector space. This way, the closeness in such a space would convey semantic relevance.

A Web crawler systematically collects information about documents from the Web to create an index of the data it is searching and it maintains an updated index through subsequent crawls. As an automatic indexer, the crawler operates in the context of listing the documents relevant to a subject or topic which one would expect in a typical user search query. Traditional general purpose Web crawlers are not easily scalable since the Web is not under the control of one operator or proprietary. They also may not be set to target specific *topics* for accurate indexing, and lag behind in time and updates to manage updated indexes/indices of the whole Web to stay current because of the distribution, subject and volume involved. To overcome these shortcomings, focused crawlers are intended to rely on the linked structure of the Web in order to identify and harvest *topically relevant* pages to increase their performance in terms of accuracy, currency and speed. A significant benefit in using focused crawlers is the possibility of decentralizing the resources and storage indexes.

There are two major open problems in focused crawling: the first problem is the prediction of the topical focus of an unvisited page before actually attempting to download the content of the page. As one of its fundamental tasks, the crawler should use specific algorithms in order to make this prediction with the highest possible accuracy. Typically, the focused crawlers download the whole content of the page, analyze it and make a decision on whether it is related to their topic of interest. Alternatively, some researches [4–6] show that the topical focus of a page can be predicted by analyzing the anchor text of its link in the parent page. Between these two extremes, in our research, we take into account several HTML structural elements of the parent page in addition to the anchor text. This will help improve the accuracy of the topical detection of the unvisited link. The second problem is prioritizing the links for later downloads. A proper priority score should be assigned to all the extracted URLs from a Web page. These URLs along with their scores will then be put in a queue to be downloaded later. This task of prioritization is very important in that some irrelevant pages should be visited and passed off on the way to reach another populated area of relevant pages. For prioritization, we employ a novel tree-like data structure called T-Graph (Treasure Graph) [7]. The traversal method in the

[*] Corresponding author.
E-mail addresses: seyfi@gwu.edu (A. Seyfi), whinchat2010@gmail.com (A. Patel).

http://dx.doi.org/10.1016/j.csi.2015.07.001
0920-5489/© 2015 Elsevier B.V. All rights reserved.

construction of T-Graph can be both top-down or bottom-up. Also, in our proposed system which is called the *Treasure-Crawler*, no resources from any other search engines are required, provided the T-Graph is set to be constructed top-down.

This study was conducted to determine a novel crawler's effectiveness and viability in crawling to fetch topic-specific documents. The system is designed and implemented in a way that all the components/modules have the minimum dependency on each other; hence, each can be changed or replaced while requiring the least manipulation of other parts except for some interfaces. The document standards such as HTML and XML are respected for this system. Also, HTTP and HTTPS communication protocol standards from IETF and W3C are considered, as well as the robot exclusion protocol known as "robots.txt".

1.1. The methodology

One of the main objectives of this research has been to enhance the accuracy of Web page classification, which is made possible by defining a flexible interpretation of the surrounding text, in addition to specific HTML data. The Dewey decimal classification (DDC) system is applied as a basis to classify the text into appropriate topic/subject boundaries. The other significant objective of this research has been to reach target documents in the shortest possible time. This is accomplished by reaching the T-Graph most matching node(s) with the document, and then calculating the distance of such nodes to the target level based on the number of documents to download [7]. As a result, due to a better determination of the topic boundary and significant decrease of the volume of downloaded documents in text format, this strategy helps the crawler update its indexes more pragmatically, accurately and rapidly. Based on these assumptions, we designed a new algorithm and built a prototype to exercise, test and validate it against functional and non-functional requirements. The summary results are only presented in this paper, while the actual detailed evaluation results are reported in a follow-up paper with the title of "Analysis and Evaluation of the Link and Content Based Focused Treasure-Crawler" [8].

In the rest of the current paper, we consecutively review some major Web crawlers and their reported experimental results, detail the requirements and evaluation criteria of a focused Web crawler, present the architectural design of the Treasure-Crawler, elaborate the employed methods and algorithms, describe the outcome of the carried-out tests and validations, and conclude with a summarized list of results and conclusions.

2. Background

Topical and focused crawling are two slightly different concepts, first introduced by Menczer [9] and by Chakrabarti [10], respectively. A focused crawler selects and seeks out pages that are relevant to (a) pre-defined topic(s). It systematically analyzes its crawl frontiers and tries to detect the pages that are most likely to be on the designated topic(s) of the crawl while it tries to avoid off-topic Web regions. As a result, this process brings considerable savings in different resources, such as network, computing and storage; hence, the crawl becomes more up-to-date. Usually, the topics of interest are optionally defined by keywords, categorized/classified standard lexicon entries, or by a set of exemplary documents. The major challenge of a focused Web crawler is the capability of predicting the relevance of a given page before actually crawling it. Achieving this goal requires particular intelligence for the crawler. Focused Web crawlers use this kind of intelligence to avoid irrelevant regions of the Web in order to make the task manageable. Additionally, a focused Web crawler should also pay attention to the ability to discover relevant regions which are separated by groups of irrelevant Web regions in order to achieve desirable Web coverage. A well-designed focused Web crawler should be able to stay in pre-defined topics as long as possible, while covering the Web as much as possible.

To index the Web, the first generation of crawlers relied on basic graph traversal algorithms, namely, the breadth-first or depth-first. An initial set of URLs is used as a seed set, and the algorithm recursively follows hyperlinks down to other documents. Document content is of less importance since the ultimate goal is to cover the whole Web. A focused crawler, on the other hand, explores the documents about a specific (set of) topic(s) and guides the searching process based on both the content and link structure of the Web. The main strategy is to associate a score with each unvisited link within the downloaded pages.

Best-first is basically an optimization to the breadth-first algorithm. When the unvisited links are extracted, an estimator tries to prioritize them. After being associated with a priority score, these links are inserted into a priority queue. These links are then fetched from the queue according to their assigned priority.

Panorama (*1996*) is one of the first systems that established a digital library by using the Web and made CiteSeer, the most popular scientific literature digital library and search engine, focusing on information technology and computer science. Panorama traverses the Web for on-topic documents in PDF and Postscript formats in the computer science field. To construct its topic of interest, Panorama submits relevant papers' main titles and the titles of references within the papers as separate exact phrase queries to Google Web APIs. Thus, the returned URLs form a positive example set, and examples from unrelated papers form a negative example set. Both of the two sets train a Naïve Bayes classifier, which guides the crawling process [11].

Shark Search Crawler (*1998*) tries to enter the areas where a higher population of relevant pages is observed, but stops searching in the areas with no or very little number of relevant pages.

InfoSpiders (*1999*) constitutes a population of adaptive intelligent agents. These agents use neural networks algorithms and are very advanced in distinguishing fruitful links to follow.

Context Focused Crawler (*2000*) builds upon constructing a multi-level tree of sample documents called the context graph. This algorithm, first proposed by Diligenti et al. [12], has been an active research area in the past decade.

Fig. 1 shows a context graph. For each target document (e.g. P) one such graph is constructed. As the target documents are supplied to the system in Level 0, another search engine is utilized to find the high ranking pages that contain a link to the current target document (e.g. A and B) and they are put in Layer 1. This process is then recursively repeated for each parent page until the graph reaches a desired number of levels. By convention, there is no connection between the nodes in a common layer. Also, if two (or more) documents in layer *i* (e.g. A and B) have a

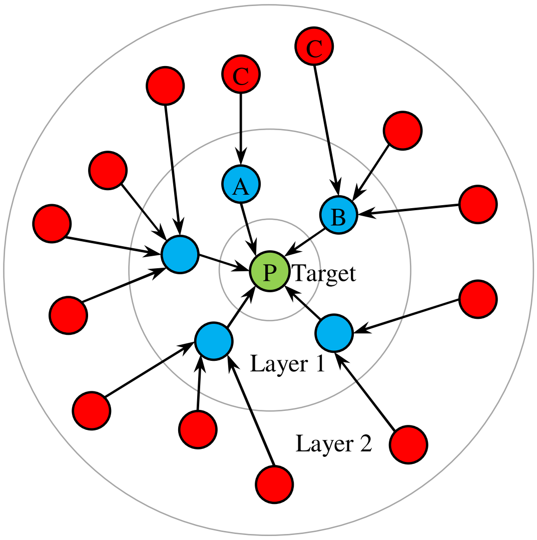

Fig. 1. A context graph.